

Proyecto

Dancistic

Nicole Lorena Vargas Patarroyo

Dancistic

Estudiante: Nicole Lorena
Vargas Patarroyo
Profesor Guía: Raimundo
Muñoz Vargas

**Agosto, 2025,
Santiago de Chile**

Memoria para optar al Título
Profesional de Diseñador en
Interacción Digital y al Grado
Académico de Licenciado en
Diseño



UNIVERSIDAD
Gabriela Mistral

Juntos escribimos tu futuro

Agradecimientos

Inicialmente quiero agradecer a mi familia, por su apoyo comprensión y ánimo constante, que me motivo a seguir adelante en los momentos de mayor dificultad y bloqueos constantes. Este logro también es de ustedes.

También quiero agradecer mis profesores y tutores, quienes me acompañaron durante este proceso con su orientación, paciencia y valiosos consejos, contribuyendo de manera significativa al desarrollo de este proyecto. Especialmente quiero agradecer a mi profesor guía Raimundo Muñoz, quien con sus consejos me ayudo a darle forma a Dancistic. También quiero agradecer enormemente al profesor Francisco Cortes, quien me ayudo con la parte fundamental de codificación, me ayudo a poner pies en tierra, me animo y mostró interés desde el inicio por mi proyecto gracias a el y su guía logre hacer lo que hoy es Dancistic.

A mis compañeros y amigos, quienes siempre me mostraron su apoyo para seguir adelante cuando me cuestionaba el “¿En que momento se me ocurrió que esto era buena idea?” Agradecer a quien inicio siendo mi compañero de clase y ahora mi pareja Thomas Castro por siempre darme animo para seguir, y brindarme las herramientas que necesitaba para poder llevar acabo mi proyecto.

Quiero agradecer de manera especial a los bailarines y participantes que me apoyaron durante el proceso, especialmente a mi grupo de baile quienes en todo momento estuvieron dispuestos a darme

consejos y posibilidades de mejora. Agradecer a todos los que se tomaron el tiempo de escucharme y llenar encuestas, y a los que, cuando ya leían y se enteraban de que trataba mi proyecto, mostraron interés por ayudarme a llevarlo acabo y hacerlo aun mas real.

Finalmente, agradezco a todas las personas y referentes que, de manera directa o indirecta, que contribuyeron a mi formación y al desarrollo de esta propuesta, la cual representa no solo un proyecto académico, sino también un paso hacia nuevas posibilidades de integrar la tecnología con la danza y la creatividad.

Gracias a la Universidad Gabriela Mistral, por ser el lugar que me acogió durante estos 4 años y medio de carrera, brindarme sus instalaciones y equipamiento para sobrellevar mi etapa Universitaria.

Y por ultimo agradecer a aquellos profesores quienes aparte de contribuir a mi formación académica también fueron un apoyo emocional durante el proceso, por que fueron mas que solo profesores en un aula, fueron amigos con toda la experiencia que yo no tenia, gracias por sentarse y escucharme cuando lo necesitaba. Profesor Rubén, profesora Carolina, profesor Eduardo, profesor Juan y directora Katrin, gracias por todo ese apoyo y palabras que me brindaron a lo largo de estos años.

Indice

Contenido

Indice.....	4	Protocolo de Uso.....	35
Introducción.....	5	Resultados e Impacto Esperados.....	37
Motivaciones.....	7	Observaciones de Testeos.....	37
Tema del proyecto.....	8	Re-definiciones.....	38
Importancia y Aporte.....	8	Definición de Parámetros.....	38
Inspiración y Referentes.....	10	Proyecciones.....	39
Investigación Creativa.....	13	Explicación del Código.....	40
Territorio, contexto y escenarios sistemicos.....	13	Conclusiones.....	61
Problemática.....	19	Planificación y Proyección Nuevas Etapas de Desarrollo.....	62
Oportunidad de diseño.....	19	Bibliografía.....	62
Usuario.....	20	Anexos.....	63
Metodología de Investigación.....	20		
Marco Conceptual.....	22		
Objetivo General.....	24		
Objetivos Específicos.....	24		
Enunciado.....	25		
Fundamentos Conceptuales.....	25		
Desarrollo Visual y Sistema de Diseño.....	25		
Atributos y Cualidades.....	26		
Definición del Sistema Interactivo.....	27		
Definición del Sistema Interactivo.....	28		
Tecnologías y Especificaciones Clave.....	30		
Iteraciones y Validaciones.....	30		
Proyección y Escalabilidad.....	32		
Modelo de Negocios.....	32		
Definición Línea Gráfica Interfaz.....	33		
Implementación.....	35		

Introducción

El presente documento expone de manera detallada el proceso de desarrollo, conceptualización y validación de Dancistic, un proyecto de título que nace desde la observación crítica del estado actual de las herramientas digitales existentes en la actualidad enfocadas en el aprendizaje y perfeccionamiento de habilidades en la danza, esto por medio de una investigación y una propuesta que combina creatividad con tecnología, esta memoria muestra el proceso de diseño e implementación de una solución interactiva centrada, inicialmente, para bailarines K-pop.

Esto en un contexto donde la tecnología avanza rápidamente en múltiples disciplinas, la danza, especialmente en su ámbito amateur y formativo, continúa enfrentando importantes carencias en cuanto a plataformas interactivas que acompañen, guíen y fortalezcan el proceso de aprendizaje de los bailarines, es aquí, en donde encontramos una de las principales motivaciones para diseñar una propuesta innovadora que aborda dicha problemática desde un enfoque funcional, creativo y técnicamente viable. Dancistic nace con el objetivo de dar respuesta a la brecha tecno-

lógica, proponiendo una plataforma innovadora y accesible que potencie la técnica, coordinación y confianza de los bailarines, fortaleciendo sus habilidades y generando una experiencia de aprendizaje más didáctica y eficiente.

Inicialmente dirigida a bailarines K-pop, una comunidad dentro del baile que destaca por su dinamismo, auto formación y fuerte presencia en redes sociales, la plataforma busca potenciar la técnica individual, la coordinación grupal, y, de manera transversal, la autoconfianza de quienes la utilizan, todo esto por medio de un sistema de retroalimentación clara, objetiva y accesible, que combina elementos visuales, narrativos y métricos para facilitar todo el proceso de aprendizaje y mejora.



Imagenes Propias

Motivaciones

Desde muy joven he sentido una profunda conexión y pasión por la danza, no solo como una manifestación artística, sino como un lenguaje propio que me ha permitido expresarme, conectar con otras personas y descubrir aspectos de mí mismo que de otra manera habrían quedado ocultos, así mismo la danza ha sido para mí un refugio y, al mismo tiempo, un desafío constante que me ha impulsado a superar mis límites y a confiar en mis propias capacidades.

Actualmente soy bailarina K-pop, aun que en ocasiones puedo bailar cualquier otro estilo, el K-pop es mi fuerte. Soy parte de la comunidad Dance Cover, quienes nos enfocamos en replicar las coreografías originales de los grupos de esta industria. Nosotros como comunidad mayormente solemos ensayar en espacios públicos, especialmente en lugares en donde nos podamos reflejar.

En cada ensayo y presentación he podido experimentar la alegría de compartir con otros esta pasión, pero también he sido consciente de las dificultades que muchos bailarines enfrentamos al

momento de perfeccionar nuestra técnica, mantener la coordinación y encontrar una retroalimentación constante, esta experiencia personal fue lo que me motivó a pensar en cómo la tecnología podía convertirse en un aliado dentro de este proceso.

Si bien la danza siempre ha sido un arte íntegramente humano, descubrí que las herramientas digitales no necesariamente la alejan de su esencia, sino que también pueden abrir nuevos caminos de aprendizaje, acompañamiento y exploración.

Es así como Dancistic surge desde mi propio recorrido como bailarina y desde el deseo de poder aportar algo notable y relevante para la comunidad, una plataforma que acompañe a otros en su travesía, que brinde apoyo para aquellos que entrenan solos, y que transforme el aprendizaje y mejora de habilidades en algo más accesible y motivador para quienes recién quieren empezar en este mundo o para quienes aun que ya pertenezcan necesitan de un apoyo extra para seguir mejorando.

Tema del proyecto

Dancistic es una propuesta innovadora que busca cambiar la forma en que los bailarines - inicialmente los bailarines K-pop - aprenden y mejoran sus habilidades, a través de tecnologías de reconocimiento y análisis de movimiento como MediaPipe, esta plataforma ofrecerá un enfoque dual: por un lado brinda un aprendizaje progresivo de las coreografías con retroalimentación en tiempo real; y por otro lado, facilita la visualización de la coordinación y sincronización grupal al comparar las ejecuciones con la coreografía original.

Importancia y Aporte

Este proyecto busca resolver la carencia de herramientas digitales que se especialicen en fortalecer la técnica, confianza y creatividad de los bailarines, además, busca fomentar la comunidad y el intercambio de conocimientos entre los usuarios, generando una experiencia más inmersiva y motivadora, tal como señala López (2021) que “las tecnologías aplicadas a la danza pueden potenciar el aprendizaje y ofrecer nuevas formas de interacción para el desarrollo artístico”

Con Dancistic propongo unir la tecnología y el diseño para abordar una necesidad específica como lo es la falta de herramientas digitales centradas en bailarines, en línea con lo expuesto por Martínez y González (2019), quienes afirman que “la integración

de recursos tecnológicos en la práctica dancística favorece la autogestión y el análisis del desempeño”.

El diseño también juega un rol importante dentro de esta propuesta ya que permite crear una interfaz intuitiva, accesible y atractiva para los usuarios, es en donde el diseño se convierte en el puente entre la innovación tecnológica y la experiencia real de los bailarines, como menciona Pérez (2020) al indicar que “el diseño de interfaces debe considerar la usabilidad y la experiencia sensorial del usuario para favorecer el aprendizaje motor”, dando a lugar un producto que no solo soluciona un problema técnico, sino que también acompaña a los usuarios en su proceso de aprendizaje.



Imagenes Propias



Inspiración y Referentes

La inspiración y referente primordial de este proyecto es el videojuego Dance Central de Xbox Kinect, Dancistic propone un sistema que no solo ayudará a mejorar las habilidades técnicas de los bailarines, sino que también ofrece un modelo de negocio accesible y adaptable para cada usuario, con opciones de suscripción, contenido gratuito rotativo y personalización de la experiencia.

En la actualidad, el mercado cuenta con diversas aplicaciones de baile y entrenamiento, tales como Dance Central, Just Dance, Arrange Us y Steezy, entre otras, así estas herramientas han ganado popularidad por su capacidad de motivar a los usuarios a través de la gamificación, el entretenimiento y la promoción de un estilo de vida activo.

Sin embargo, la mayoría de estas plataformas están enfocadas principalmente en el aspecto lúdico o en el fitness general, dejando en segundo plano el desarrollo técnico y la mejora específica de habilidades evidencia una oportunidad para propuestas como

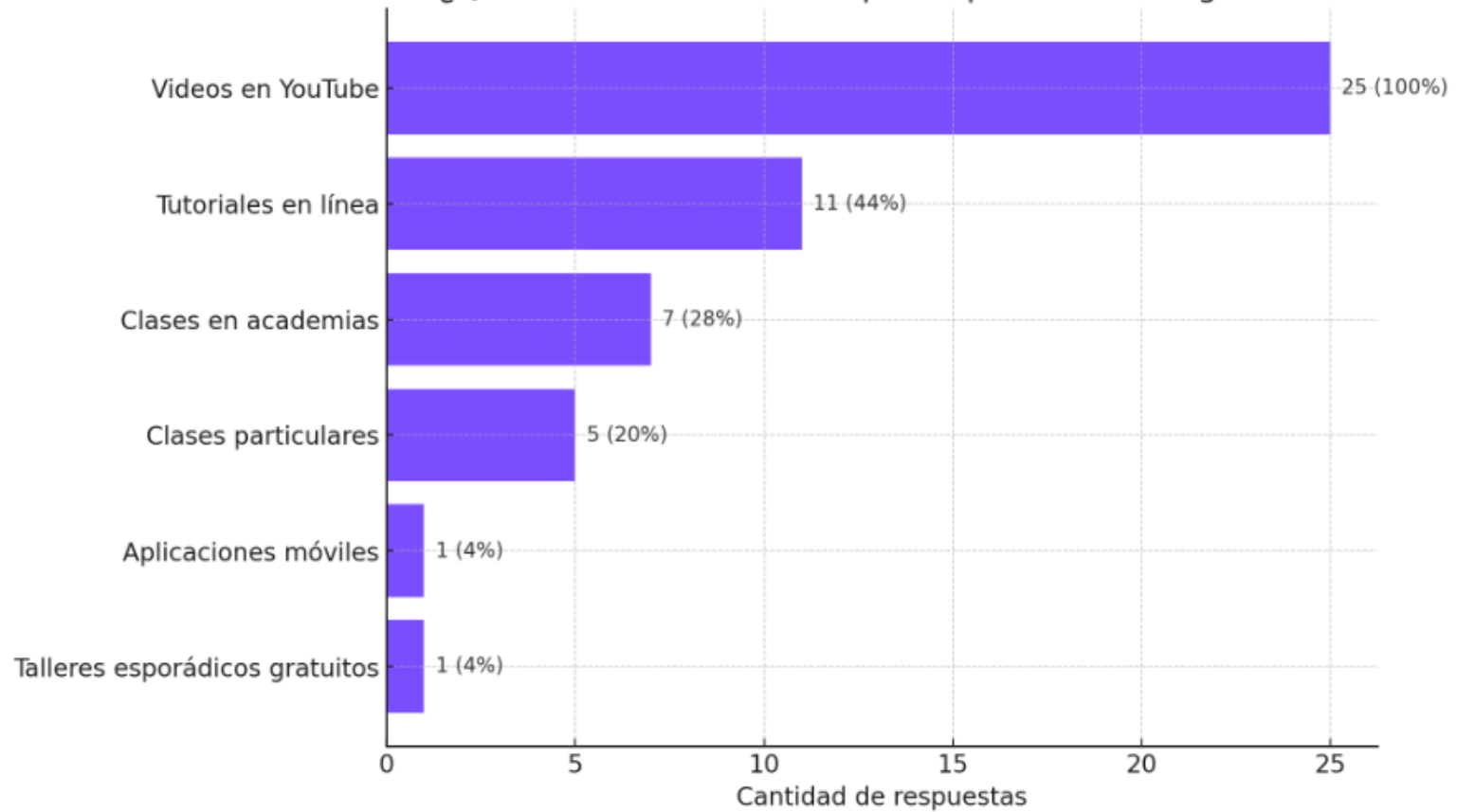
Dancistic, que, a diferencia de estos referentes, busca centrarse en el aprendizaje estructurado y en la retroalimentación detallada, ofreciendo a los usuarios una experiencia orientada a la progresión y perfeccionamiento de sus capacidades.

En el desarrollo de Dancistic, se tomaron como referencia ciertos elementos positivos de estos productos existentes: De Dance Central y Just Dance, se rescató la capacidad de motivar a través de la interacción y el ritmo, mientras que de Steezy se valoró la estructura visual y las herramientas de práctica que facilitan la repetición de movimientos. No obstante, el enfoque del proyecto se diferencia al priorizar el análisis técnico de las coreografías y la entrega de retroalimentación personalizada, combinando lo mejor del entretenimiento con la precisión del entrenamiento guiado.



Imagen ilustrativa del Juego Dance Central

¿Qué recurso sueles usar para aprender coreografías?



Graficos de encuestas realizadas

Investigación Creativa

El desarrollo de Dancistic surge a partir de mi propia experiencia y mi pasión por la danza, de un análisis de las herramientas digitales actualmente disponibles para los bailarines. Para entender las necesidades reales de los usuarios, realice encuestas y entrevistas con bailarines K-pop, usuarios jóvenes con un interés profundo en perfeccionar sus habilidades dentro del baile.

Los resultados de la investigación muestran que más del 70% de los usuarios consideran el baile como un hobby importante en sus vidas, pero enfrentan diversos desafíos: desde la falta de confianza y espacio adecuados para la práctica, hasta problemas de coordinación grupal en grupos de más de 4 personas. Además la mayoría de los entrevistados manifestó no tener conocimientos acerca de herramientas digitales para mejorar su técnica, recurriendo principalmente a Youtube como fuente de aprendizaje.

La investigación también permitió perfilar distintos tipos de usuarios: desde una joven aprendiz que busca ganar confianza, hasta un bailarín profesional que busca perfeccionar sus movimientos y ayudar a otros. Estos perfiles ayudaron a definir la experiencia que se le da al usuario y la interfaz de la plataforma, asegurando que la propuesta cumpla tanto con las necesidades prácticas como los deseos de los usuarios.

Así, la investigación creativa fue clave para establecer los cimientos conceptuales de Dancistic y delinear una propuesta que no solo innove en la forma de aprender sino que sea accesible, motivadora y adaptable a la diversidad de bailarines.

Territorio, contexto y escenarios sistemicos

Dancistic se desarrolla dentro del contexto global de la cultura K-pop, un fenómeno cultural y social que se ha expandido globalmente y que, para los jóvenes Chilenos y de Latinoamérica, se ha convertido en un medio de expresión, entretenimiento y comunidad, tal como señalan Kim y Park (2021) al afirmar que “el K-pop se ha convertido en un fenómeno transnacional que articula identidades juveniles y fomenta la creación de comunidades globales en línea y presenciales”. Esta cultura no solo abarca la música sino también un universo de prácticas como lo son los dance covers, eventos temáticos y competencias, donde la precisión y coordinación de las coreografías son elementos fundamentales, esto coincide con lo descrito por López (2022), quien explica que “la fidelidad en la ejecución coreográfica es un aspecto central en la cultura del K-pop y en la evaluación de los dance covers”.

El entorno en el que Dancistic surge está marcado por una generación que combina lo físico y lo digital en su día a día: jóvenes que buscan en las plataformas tecnológicas no solo información y aprendizaje, sino también la validación y conexión con otros. Según el estudio de Martínez (2019), “las redes sociales y plataformas de vídeo son espacios clave para la socialización, la visibilización y el aprendizaje autónomo de coreografías en jóvenes latinoamericanos”. Gracias al auge de plataformas como YouTube y Tiktok los jóvenes emprenden en el mundo de la danza aprendiendo coreografías por imitación, para cumplir un trend o challenge, pero a menudo de algo tan básico como un trend de 10 segundos, nace la pasión por el baile, y por aprenderse coreografías más extensas y complejas.

Actualmente existe una brecha clara en la disponibilidad de herramientas digitales diseñadas específicamente para apoyar a los bailarines en su proceso de aprendizaje y perfeccionamiento, especialmente en el ámbito del K-pop. Este vacío es más evidente para quienes tienen un nivel básico e intermedio, quienes enfrentan barreras como la falta de espacios adecuados, la inseguridad o la falta de orientación profesional, como advierte Pérez (2020) al señalar que “la ausencia de entornos de aprendizaje adaptados a principiantes dificulta el progreso técnico y la permanencia en la práctica”.

En este escenario, Dancistic propone una solución que no solo aborda la necesidad técnica de mejorar la ejecución y coordinación de los bailarines, sino que también reconoce que hay un valor cultural y social de la danza como un espacio de identidad y comunidad. Esto se alinea con lo planteado por Castro y Ramírez (2018), quienes sostienen que “la danza es, además de una práctica artística, un espacio de construcción social y de fortalecimiento de la identidad colectiva”. El proyecto se sitúa en un entorno donde el diseño de experiencias digitales puede ser un puente entre la pasión y el acceso a herramientas que potencien su desarrollo personal y artístico.

¿Qué hacen los bailarines actualmente?

Si bien existen academias, instructores y/o profesores que enseñan técnicas de baile, la mayoría de los usuarios no tienen acceso a estos, ya sea por condiciones económicas o geográficas. Entonces, cómo es su proceso de aprendizaje y mejora?

El proceso de un bailarín para aprender una coreografía, mejorar su ejecución y llegar a una buena coordinación grupal, en general suele ser largo, poco estructurado y en la mayoría de los ca-

sos dependiente de herramientas no diseñadas especialmente para este fin. A través de encuestas y visualización en el territorio o lugares de ensayo de los bailarines, y aparte basándose en experiencias personales, es posible reconstruir una secuencia de acciones que refleja la experiencia que viven los usuarios de manera habitual. Este proceso aparte de tener múltiples etapas, logra poner en evidencia las barreras, limitaciones y necesidades que surgen a lo largo de su experiencia de aprendizaje y mejora.

En primera instancia se debe tener un interés sobre una coreografía, los bailarines suelen tener varios motivos para elegir bailar una coreografía, estos van desde un gusto particular por el grupo que baila originalmente la coreografía, también por el interés neto de querer aprenderla sin necesidad de conocer al grupo, también se dan casos en los que se la aprenden por ayudar a su grupo de amigos a llevar la coreografía a término, o incluso solo para poder ser un bailarín de apoyo dentro de la presentación final de un grupo dance cover. En esta fase el interés suele estar vinculado a una necesidad de expresión personal, integración social, sentido de pertenencia grupal o admiración por un grupo musical en particular.

Una vez identificada la coreografía que se desee aprender, los bailarines recurren en la mayoría de los casos a Youtube para buscar el vídeo de la coreografía, el cual vendría siendo el vídeo de referencia por el cual deben partir, después identifican a que bailarín desean hacer, ya que, las coreografías de K-pop suelen ser interpretadas por los mismos artistas y cada artista está dentro de la coreografía, el bailarín debe escoger a quien le interesa más hacer.

Después de esto empieza el proceso de aprendizaje neto, en general no se buscan tutoriales paso a paso de la coreografía ya

que con estados existe la problemática de la generalización de toda la coreografía, enseñando una versión más generalizada en donde solo se toma en cuenta la coreografía que hace el que esté en el medio, esto suele ser un problema por que, como se mencionó anteriormente, cada bailarín hace un artista distinto, y normalmente la coreografía suele variar entre integrantes. Entonces, que vídeo o vídeos usan para aprenderse una coreografía? suelen buscar inicialmente el vídeo dance practice (vídeo de práctica de coreografía) del grupo original, este en la mayoría de los casos buscan que sea tipo mirror (tipo espejo) para que les sea más fácil imitar los movimientos y los pasos.

En ocasiones, solo el dance practice no suele ser suficiente, ya que a veces no se logran visualizar algunos de los pasos, así que, recurren a la búsqueda de presentaciones del grupo en distintos ángulos o fancams (vídeos en donde se sigue a solo uno de los integrantes de la coreografía) estos facilitan la imitación directa del bailarín, por que si, los dance cover tratan de la imitación de coreografías y expresiones de un grupo K-pop en particular. Este material se convierte en la parte principal y primordial para la fase de memorización, pero también presenta desafíos importantes como la falta de marcación de tiempos, que para bailarines que recién están empezando, suele ser difícil sacar, otro de los desafíos más comunes es el ángulo de la cámara o velocidad de los movimientos que pueden dificultar la comprensión total de una secuencia de baile.

Para lograr un aprendizaje efectivo de la coreografía es necesaria la combinación de elementos visuales y una estructura secuencial bien definida. Los bailarines reproducen vídeo una y otra vez, desaceleran la reproducción, retroceden segmentos y dividen la coreografía en partes manejables que se puedan asimilar de forma gradual. En este contexto es común que el usuario intente replicar los pasos frente a un espejo o cualquier superficie que le permita reflejarse, en ausencia de este, directamente lo hacen frente a la cámara de su teléfono móvil.

Esta etapa requiere una gran cantidad de tiempo, concentración y repetición, y es también donde empiezan a aparecer las primeras frustraciones: la imposibilidad de identificar que se está haciendo mal o que parte específica necesita mejorar.

Ante la falta de retroalimentación externa, muchos bailarines implementan estrategias de auto evaluación. Una de las más frecuentes consiste en grabarse ejecutando la coreografía o parte de ella, luego, comparan esa grabación con el vídeo original de referencia, intentando detectar errores de ejecución, de-sincronización o falta de precisión en los movimientos, es así como esta comparación se realiza de forma paralela, reproduciendo ambos vídeos al mismo tiempo desde dos ventanas distintas o en dispositivos diferentes, aunque, en algunos casos se llega a recurrir a aplicaciones de edición básica como CapCut o InShot para alinear las tomas y observar diferencias cuadro a cuadro.

Este método aunque funcional hasta cierto punto, presenta algunas limitaciones significativas. En primer lugar, la comparación depende de la capacidad visual y analítica del bailarín, lo que conduce a una carga subjetiva alta y puede derivar en interpretaciones incorrectas, por otro lado, el proceso es manual, lento y requiere de múltiples repeticiones, lo que desincentiva la práctica sostenida y además en contextos grupales, donde el objetivo es evaluar la coordinación entre integrantes, se vuelve particularmente complejo de identificar con claridad quien está desfasado o descoordinado, que parte de la coreografía necesita ajuste o como sincronizar los tiempos de manera precisa.

Superada esta etapa inicial, y una vez que el bailarín considera haber dominado la secuencia individualmente, se da paso a los ensayos grupales, esta fase se caracteriza por la búsqueda

de uniformidad y coordinación entre todos los integrantes del grupo, en donde, el primer desafío a enfrentar, es la unificación de las coreografías aprendidas individualmente, el ver si todos vieron los mismos movimientos o si alguno se saltó algún paso que por sí solo no logro ver, después de esto, nuevamente se recurre a la grabación como única herramienta de evaluación, el grupo se graba bailando la coreografía y posteriormente revisa los vídeos para observar y hay coherencia en los movimientos, si las transiciones son limpias y si los tiempos están correctamente sincronizados, esta observación también se realiza sin ayuda tecnológica especializada, limitándose a lo que el grupo puede detectar a simple vista.

En este punto, surgen varios desafíos adicionales, la crítica entre pares puede generar tensiones, especialmente si no existen criterios claros para evaluar la ejecución, asimismo, la falta de precisión en la observación hace que muchas veces los errores se mantengan o se ignora, la ausencia de indicadores objetivos de rendimiento dificulta la mejora progresiva y puede demorar el avance general del grupo.

Finalmente, una vez que el grupo siente que ha alcanzado un nivel adecuado de ejecución, se orienta hacia la preparación de una presentación o grabación final, esta puede tener como objetivo un evento presencia, una competencia o simplemente la publicación del vídeo en redes sociales. No obstante, incluso en esta fase final, la evaluación del rendimiento sigue dependiendo exclusivamente del juicio visual colectivo, sin métricas claras ni retroalimentación segmentada.

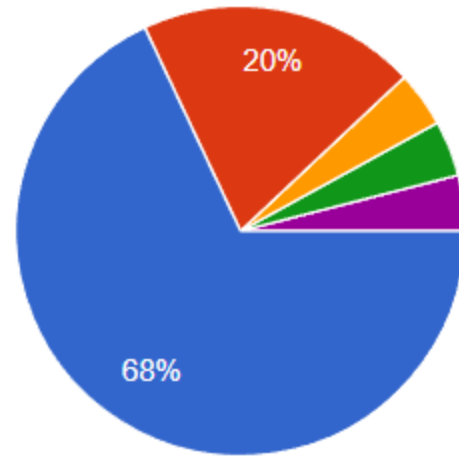
En conclusión con lo anterior, lo que viene siendo el User Journey, actualmente los procesos de aprendizaje y coordinación grupal está fuertemente condicionado por la falta de herramientas di-

señadas específicamente para apoyar estas tareas, se trata de un recorrido manual, artesanal y autodidacta, que depende del ensayo y error, del uso creativo de plataformas generalistas y de la motivación personal del usuario.

Este escenario pone de manifiesto una serie de carencias: ausencia de retroalimentación objetiva, dificultad para realizar comparaciones precisas, escasa capacidad de medición del progreso, y una clara desconexión entre la práctica individual y la evaluación grupal y, si bien los bailarines han logrado construir estrategias funcionales dentro de sus posibilidades, estas están lejos de representar un sistema óptimo o eficiente de aprendizaje, la implementación de soluciones digitales especializadas podría no sólo acortar los tiempos de mejora, sino también ofrecer una experiencia más enriquecedora, clara y motivadora para todos los niveles de usuarios.

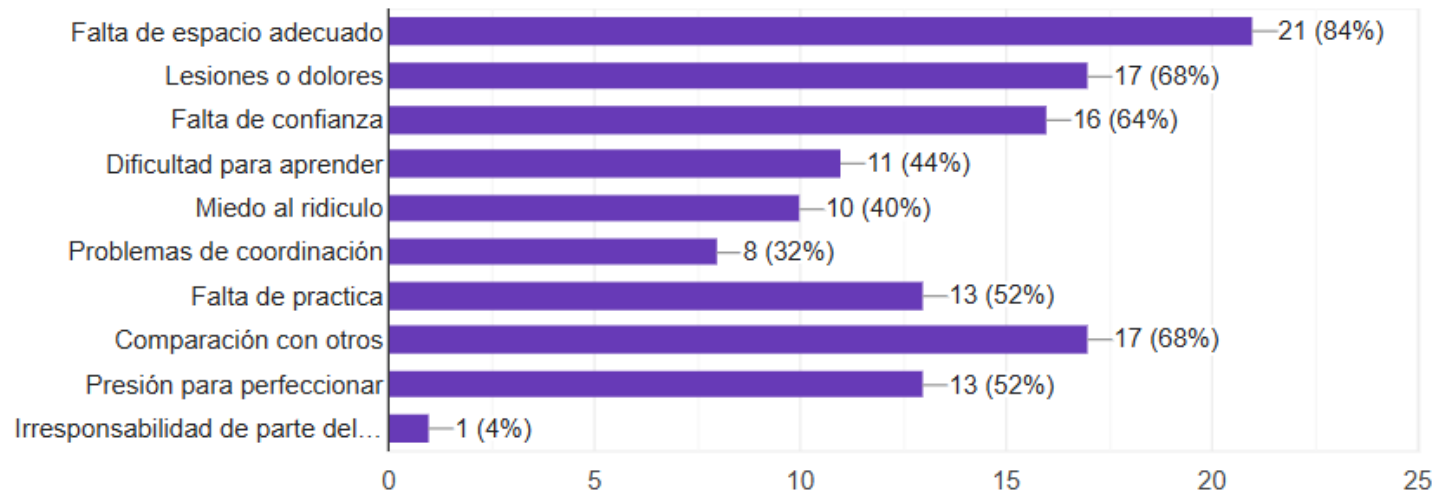


Imágenes Propias



- Un hobby
- Mi profesión
- Hobby que amo pero también es obligatorio por mi salud y enfermedad crónica
- proxima profesion
- Es algo que me gusta mucho hacer pero que por ahora no puedo hacer que sea mi profesión

Graficos de encuestas realizadas



Problemática

En la actualidad, los jóvenes bailarines de nivel básico e intermedio enfrentan importantes desafíos para aprender coreografías y mejorar sus habilidades, tanto individuales como grupales, especialmente fuera de un entorno presencial.

La enseñanza tradicional de la danza depende en gran medida de la observación y corrección directa de un instructor, lo que limita la posibilidad de practicar de forma autónoma y recibir retroalimentación técnica oportuna, es así como esta situación provoca que errores en la ejecución, la coordinación o la técnica se mantengan en el tiempo, afectando el progreso y la motivación, si bien, existen herramientas digitales para el consumo de contenido o entrenamientos generales, estas no suelen ofrecer un análisis detallado de los movimientos ni evaluaciones personalizadas adaptadas a las necesidades específicas de los bailarines, es entonces cuando la ausencia de soluciones interactivas que integren tecnología para apoyar el aprendizaje autónomo y colaborativo dificulta que el proceso formativo sea más eficiente, accesible y motivador para este grupo de usuarios.

Es aquí cuando se determina que la problemática a abordar sería: La dificultad en los jóvenes bailarines de nivel básico e intermedio para aprender coreografías y mejorar sus habilidades individuales y/ o grupales.

Oportunidad de diseño

Existe la posibilidad de desarrollar una solución interactiva que ayude a jóvenes bailarines con conocimientos básicos e intermedios a practicar y perfeccionar su técnica desde cualquier lugar, sin depender por completo de la presencia física de un instructor, gracias a tecnologías como el reconocimiento de pose y el análisis de movimiento, se puede ofrecer retroalimentación visual y cuantitativa en tiempo real, detectando errores, reforzando aciertos y guiando el progreso de forma personalizada, esto no solo facilita el aprendizaje individual, sino que también impulsará el trabajo en grupo, optimizando la coordinación y sincronización en coreografías colectivas.

Aquí es en donde se propone una plataforma digital, que reúna tanto la retroalimentación dada por profesionales - esto por medio de la red social incorporada- como la retroalimentación en tiempo real que se obtiene mediante el reconocimiento de pose y el análisis de movimiento, con este enfoque podría democratizar el acceso a la formación en danza, adaptarse a distintos niveles de habilidad y mantener la motivación a través de objetivos claros, métricas de progreso y recursos didácticos atractivos.

Usuario

El proyecto está dirigido a personas jóvenes, entre 14 y 25 años, con conocimientos básicos o intermedios en danza, que buscan mejorar sus habilidades técnicas, aprender nuevas coreografías y optimizar su desempeño individual o grupal. Dentro de este público objetivo, se identifican tres perfiles principales que representan distintas necesidades y motivaciones de uso:

El aprendiz: Persona que recién comienza a adentrarse en el mundo de la danza. Busca aprender fundamentos básicos, adquirir confianza en sus movimientos y progresar a un ritmo accesible. Valora especialmente la retroalimentación clara y sencilla, así como las guías visuales paso a paso que le permitan mejorar desde la práctica individual.

El bailarín grupal: Forma parte de un equipo o agrupación y su objetivo principal es perfeccionar la coordinación y sincronización del grupo para lograr presentaciones más limpias y efectivas, especialmente en contextos de competencias. Necesita herramientas que permitan evaluar la ejecución colectiva, detectar desajustes y reforzar la cohesión del conjunto.

El bailarín profesional: Posee habilidades ya desarrolladas y experiencia previa, pero busca perfeccionarse constantemente y explorar nuevas técnicas. Además, tiene interés en ayudar a otros, ya sea como mentor o colaborador, por lo que valora funciones que faciliten la enseñanza, el análisis detallado de movimientos y la posibilidad de compartir conocimientos con la comunidad.

Metodología de Investigación

Para comprender en profundidad las necesidades, expectativas y problemáticas de los usuarios, se aplicó una metodología mixta que combinó encuestas y entrevistas semiestructuradas.

Encuestas

Las encuestas se realizaron mediante Google Forms y fueron distribuidas a través de redes sociales, ya que este medio ofrecía el mayor alcance y rapidez para obtener respuestas. Se indicó que podían participar personas de cualquier edad y con cualquier nivel de experiencia en baile, con el objetivo de identificar el rango etario de los potenciales usuarios, su nivel de experiencia, motivaciones y los géneros que practican.

En total se recibieron 25 respuestas. Los principales hallazgos fueron:

- **Importancia del baile como actividad personal:** Más del 70% declaró que el baile es un pasatiempo significativo, y al menos un participante lo considera su profesión. Esto refuerza la idea de que los bailarines aficionados son un público ideal para soluciones accesibles, sin altos requerimientos técnicos o económicos.
- **Perfil predominante:** La mayoría practica baile K-pop, lo que orienta la fase inicial del proyecto a este nicho.
- **Barreras principales:** Falta de tiempo y recursos económicos, lo que refuerza la necesidad de una herramienta económica, accesible y que optimice el tiempo de práctica.
- **Motivaciones clave:** Expresión artística, diversión, terapia y

preparación para eventos, lo que sugiere que la plataforma debe combinar feedback técnico con un enfoque motivador.

- **Desafíos comunes:** Falta de espacios de práctica, baja confianza, dificultad de aprendizaje, lesiones y comparaciones con otros.
- **Recursos y hábitos actuales:** Uso de YouTube como principal herramienta de aprendizaje, complementado con auto evaluación a través de grabaciones y el apoyo entre compañeros.
- **Expectativas tecnológicas:** Los participantes mencionaron ideas como “una IA que corrija los pasos”, “una app que ayude con la coordinación y las posiciones” y “una que permita ver tiempos y posiciones en grupo”.

Estos datos validan la necesidad de una herramienta que ofrezca retroalimentación visual, técnica y motivadora, con un modelo económico accesible como el freemium, que combine funciones gratuitas con opciones de pago para características avanzadas.

Entrevistas

Se realizaron entrevistas semiestructuradas a cinco usuarios con distintos niveles de experiencia y a un profesor de baile, con el fin de profundizar en las experiencias, necesidades y percepciones que no podían captarse completamente a través de las encuestas.

Insights principales de las entrevistas a usuarios

Barreras y desafíos

- Falta de retroalimentación en tiempo real fuera de las clases presenciales.
- Dificultad para detectar y corregir errores sin un instructor.
- Problemas para coordinarse en grupo.

- Limitaciones de espacio y recursos para practicar.
- Ejemplo: “Cuando practico sola, no me doy cuenta de que mi brazo está mal hasta que me lo corrigen en clase” [Entrevista Fernanda Vallejos] (comunicación personal, 6 de octubre 2024).

Motivaciones y objetivos

- Mejorar técnica personal y sincronización grupal para competencias.
- Mantener un progreso visible y motivador.
- Ejemplo: “Lo que más quiero es que mi grupo esté completamente coordinado, porque eso marca la diferencia en el escenario” [Entrevista Jeremías Veloso] (comunicación personal, 6 de octubre 2024).

Uso actual de herramientas digitales

- Dependencia de YouTube para aprender y practicar.
- Uso frecuente de grabaciones propias para auto evaluación.
- Falta de conocimiento sobre herramientas especializadas para corrección técnica.
- Ejemplo: “Siempre veo el vídeo de nuevo y me fijo si lo hice igual... pero a veces no sé qué está mal” [Entrevista Andrea Figueroa] (comunicación personal, 6 de octubre 2024).

Necesidades tecnológicas

- Comparar ejecución con referencia visual clara.
- Feedback técnico y motivador.
- Evaluación tanto individual como grupal.
- Ejemplo: “Sería genial que me dijera en qué momento me atraso o me adelanto en la coreografía” [Entrevista Matilde

Osorio] (comunicación personal, 6 de octubre 2024).

- Insights principales de la entrevista al profesor de baile (Frank)

Retroalimentación personalizada

La corrección inmediata es clave para mejorar técnica y evitar lesiones.

Ejemplo: “Si el alumno repite muchas veces el mismo error sin corregirlo, no solo lo aprende mal, sino que puede lesionarse” [Entrevista Frank Flores] (comunicación personal, 25 de agosto 2024).

Aplicaciones prácticas para un docente

- Usar la herramienta como apoyo en clases y seguimiento de progreso.
- Posibilidad de estandarizar evaluaciones técnicas.

Recomendaciones pedagógicas

- Interfaz simple y visualmente clara.
- Métricas y visualizaciones fáciles de interpretar.
- Feedback adaptado al nivel del bailarín.

Con esta metodología combinada, se obtuvo una comprensión integral de las necesidades y motivaciones del público objetivo, validando la propuesta de Dancistic como una herramienta tecnológica capaz de ofrecer retroalimentación personalizada, accesible y motivadora tanto a nivel individual como grupal.

Marco Conceptual

El proyecto Dancistic se fundamenta en un marco teórico interdisciplinario que integra conceptos y teorías provenientes de la danza, el entrenamiento deportivo, la psicología del aprendizaje, la tecnología aplicada al análisis de movimiento y el diseño de experiencias interactivas, tal como indica López (2022), “el desarrollo de competencias en danza requiere un enfoque que combine la técnica corporal con estrategias pedagógicas apoyadas en recursos tecnológicos”. Este marco proporciona la base conceptual necesaria para comprender cómo la tecnología puede potenciar los procesos de aprendizaje y perfeccionamiento en jóvenes con conocimientos básicos e intermedios de danza.

La necesidad principal surge de las dificultades que enfrentan estos bailarines para acceder a retroalimentación técnica precisa y constante fuera del contexto presencial. Factores como la falta de tiempo, recursos económicos, espacios de práctica y herramientas especializadas impactan negativamente en su progreso, reduciendo la motivación y limitando las oportunidades de mejora. En este sentido, Castro y Ramírez (2018) destacan que “la continuidad en la práctica y la calidad del feedback recibido son determinantes en el avance técnico del bailarín”.

A través de la incorporación de tecnologías como el reconocimiento de pose y movimiento asistido por inteligencia artificial, sumado a un enfoque centrado en la experiencia del usuario, Dancistic busca ofrecer un espacio digital accesible, motivador y adaptado a los distintos perfiles de bailarines. Este marco conceptual permite sustentar teóricamente la pertinencia de la propuesta y conectar las necesidades detectadas con las soluciones

que ofrece la plataforma, siguiendo la línea de lo propuesto por Martínez (2019), quien afirma que “el diseño de entornos interactivos debe responder a las necesidades reales del usuario y facilitar su integración en contextos de aprendizaje formal e informal”.

Danza

La danza es una manifestación artística y cultural basada en el movimiento del cuerpo, generalmente acompañada de música, que permite la expresión de ideas, emociones o narrativas. En el contexto de este proyecto, la danza se entiende tanto como una actividad recreativa y social, como un medio de desarrollo físico y técnico.

Danza urbana

Se refiere a un conjunto de estilos de baile surgidos en entornos urbanos, caracterizados por movimientos enérgicos, creatividad y una fuerte conexión con la música popular contemporánea. Dentro de este grupo se incluye el K-pop dance cover, estilo predominante en la comunidad que inspiró este proyecto.

K-pop dance cover

Práctica en la que bailarines recrean las coreografías originales de grupos musicales coreanos, buscando precisión en movimientos, coordinación grupal y sincronización con la música. Es una disciplina que combina técnica, memoria coreográfica y expresión escénica.

Aprendizaje motor

Proceso por el cual una persona adquiere, perfecciona y automatiza habilidades físicas a través de la práctica y la retroalimentación. En la danza, este proceso implica coordinación, equilibrio, memoria muscular y control del movimiento.

Feedback (retroalimentación)

En el ámbito del aprendizaje de la danza, el feedback es la información que el bailarín recibe sobre su desempeño, con el fin de mejorar. Puede ser técnico (correcciones específicas sobre postura, tiempo, alineación) o emocional (mensajes que motivan y refuerzan la confianza).

Feedback positivo

Estrategia que busca reforzar conductas o habilidades correctas a través de mensajes motivadores y visuales, fomentando la constancia y reduciendo la frustración durante el aprendizaje.

Entrenamiento autónomo asistido

Modalidad de práctica en la que el bailarín entrena sin un instructor presente, pero recibe asistencia a través de herramientas digitales que le entregan datos, métricas y análisis sobre su desempeño.

Visión por computadora

Área de la inteligencia artificial que permite a las máquinas interpretar y procesar imágenes o vídeos para extraer información útil. En Dancistic, esta tecnología se usa para analizar el movimiento de los bailarines.

Reconocimiento de pose

Técnica que, mediante visión por computadora, identifica la posición y orientación de las articulaciones del cuerpo humano en una imagen o vídeo. Permite medir ángulos, desplazamientos y posturas.

Análisis de movimiento

Proceso de evaluación que compara los movimientos realizados por el usuario con un patrón o referencia (en este caso, la coreografía original), identificando aciertos, errores y áreas de mejora.

Inteligencia artificial aplicada a la danza

Uso de algoritmos que procesan datos de movimiento para generar retroalimentación personalizada. Esto permite adaptar la corrección y el entrenamiento a las necesidades específicas de cada bailarín.

Gamificación

Aplicación de elementos de juego (puntos, retos, niveles) en contextos no lúdicos para aumentar la motivación y la participación del usuario. En Dancistic, se usa para fomentar la práctica constante y la superación personal.

Modelo freemium

Estrategia de negocio en la que se ofrece acceso gratuito a las funciones básicas del producto, mientras que las características avanzadas requieren un pago. Esto hace que la herramienta sea accesible para un mayor número de usuarios y sostenible económicamente.

Objetivo General

Diseñar e implementar una plataforma digital interactiva que utilice tecnologías de reconocimiento de pose e inteligencia artificial para brindar retroalimentación técnica y motivacional a bailarines de nivel básico e intermedio, facilitando su aprendizaje autónomo, mejorando su técnica y promoviendo la constancia en la práctica de la danza.

Objetivos Específicos

1. Identificar las necesidades, motivaciones y dificultades de los bailarines de nivel básico e intermedio, junto con el análisis de las herramientas digitales existentes, con el fin de orientar el diseño y funcionalidad de la plataforma.
2. Desarrollar e implementar un sistema interactivo basado en reconocimiento de pose y análisis de movimiento, que ofrezca retroalimentación técnica y motivacional en tiempo real mediante métricas visuales claras y accesibles.
3. Diseñar, evaluar y optimizar una interfaz de usuario intuitiva y gamificada, validando su eficacia a través de pruebas con usuarios reales para garantizar una experiencia motivadora y funcional.

Enunciado

Dancistic se comprende como una plataforma web interactiva orientada al aprendizaje progresivo y a la mejora objetiva del desempeño coreográfico. Mediante el procesamiento automático de vídeo, la aplicación entrega retroalimentación cuantificable sobre la ejecución individual y la coordinación grupal de los bailarines, destacando con precisión las áreas corporales que requieren ajuste (cabeza, brazos, torso y piernas) y señalando, con marcas temporales, los instantes específicos en que se producen errores.

Fundamentos Conceptuales

Aprendizaje asistido por visión computacional

- Uso de MediaPipe Pose para el rastreo de 33 landmarks corporales a 30fps.
- Normalización de coordenadas y penalización progresiva de la desviación angular para generar porcentajes de similitud.

Retroalimentación multimodal

1. Visual: Superposición de gráficos en verde, amarillo y rojo sobre las partes del cuerpo según grado de acierto.
2. Narrativa: Reporte textual por zona corporal que describe nivel de logro y lista de segundos donde se concentran los errores.
3. Métrica: Indicador global de desempeño y porcentajes parciales para brazos, piernas, cabeza y torso.

Iteración incremental

Ciclo “Intento – análisis – corrección – nuevo intento” con registro histórico para que el usuario observe su progreso.

Gamificación ligera

Sistema de créditos obtenidos al superar umbrales de similitud o ver anuncios, fomentando la práctica repetida.

Desarrollo Visual y Sistema de Diseño

Identidad visual:

El proyecto Dancistic utiliza una identidad visual que combina dinamismo, modernidad y un toque lúdico, buscando representar la energía y diversidad del mundo de la danza.

Logotipo:

Diseñado con la tipografía Alstoria, que aporta personalidad, elegancia y un carácter distintivo. Sus formas redondeadas y detalles únicos transmiten cercanía y creatividad, elementos clave en la propuesta del proyecto.

Dancistic

Tipografía de la Plataforma:

Se emplea Mukta Mahee para los textos de la interfaz y contenidos, por su excelente legibilidad en entornos digitales y su versatilidad para adaptarse a distintas jerarquías visuales.

Paleta de Colores:

#EB67F2 – Rosa vibrante que transmite creatividad y energía.

#4DA2F4 – Azul medio que aporta frescura y calma, equilibrando la composición visual.

#13F2E9 – Cian brillante que refuerza la idea de tecnología y dinamismo.

#FDA7F0 – Rosa suave que complementa la identidad con un matiz amigable y accesible.

Gradiente – Transición armónica entre rosa y azul, representando la conexión entre emoción (rosa) y precisión técnica (azul).

Justificación de la elección:

La identidad visual se construyó considerando dos ejes conceptuales:



- Tecnología y precisión – Representada por los tonos fríos (azules y cian) que evocan confianza y profesionalismo en el análisis de movimiento.
- Creatividad y pasión por la danza – Representada por los tonos cálidos y vibrantes (rosas) que transmiten energía, expresividad y conexión humana.

El contraste entre colores vivos y neutros facilita la atención en los elementos clave de la interfaz, mientras que la tipografía seleccionada equilibra estética y funcionalidad, garantizando una experiencia visual clara y atractiva.

Atributos y Cualidades

Objetivo vinculado	Atributo de la solución	Mecanismo concreto
Potenciar técnica individual	Precisión de feedback	Desglose por parte corporal con tolerancia graduada (0,07 \times 0,20 rad).
Mejorar coordinación grupal	Similitud multitrack	Comparación sincronizada de varios bailarines en un mismo vídeo.
Reducir tiempo de aprendizaje	Flujo ágil de carga-análisis	Procesamiento local < 15 s por minuto de vídeo (MediaPipe + FFmpeg).
Incrementar confianza	Mensajería motivacional	Mensajes positivos cuando el avance supera hitos de 5 %.

Definición del Sistema Interactivo

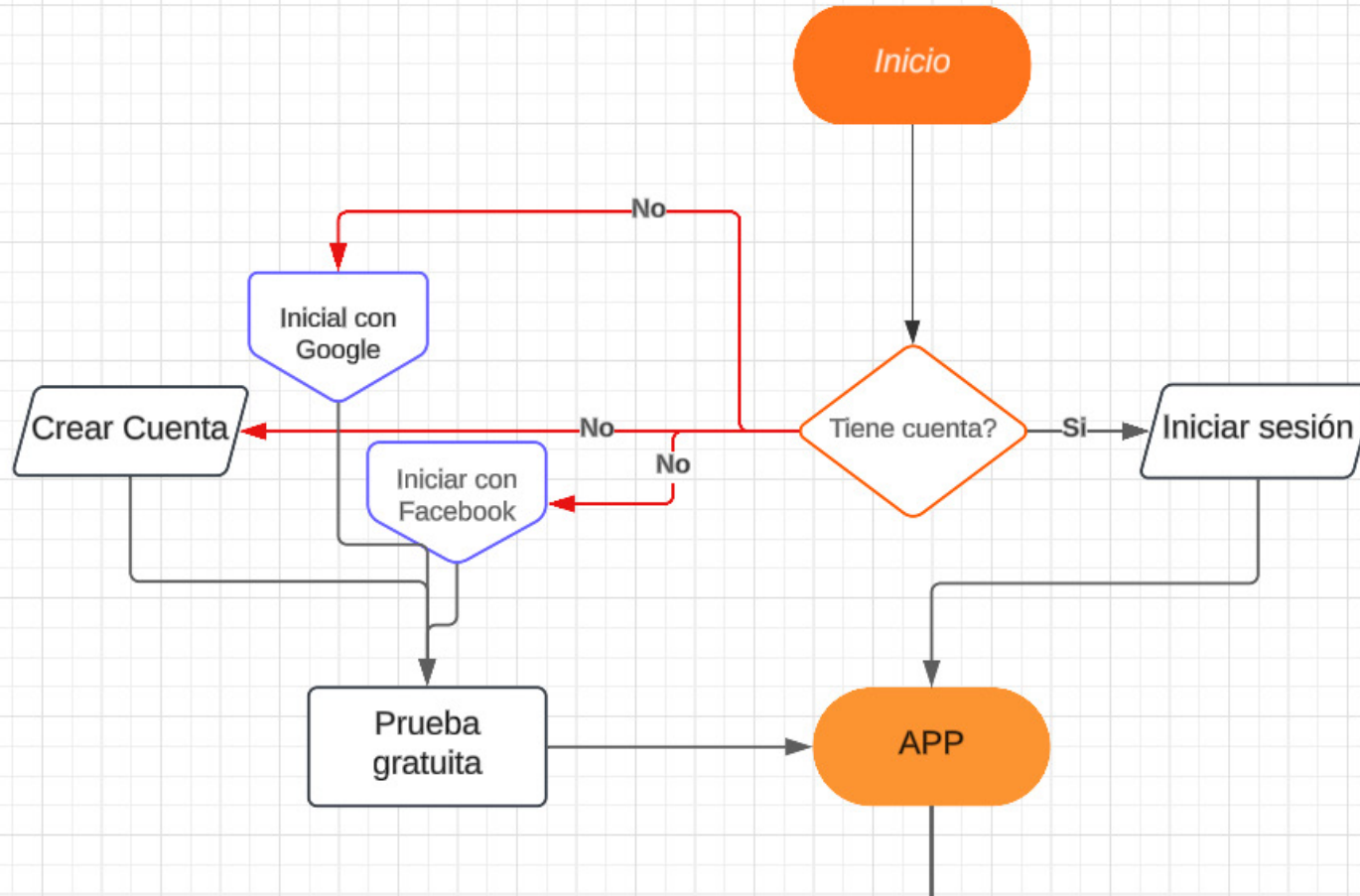


Imagen del Diagrama de flujo

Definición del Sistema Interactivo

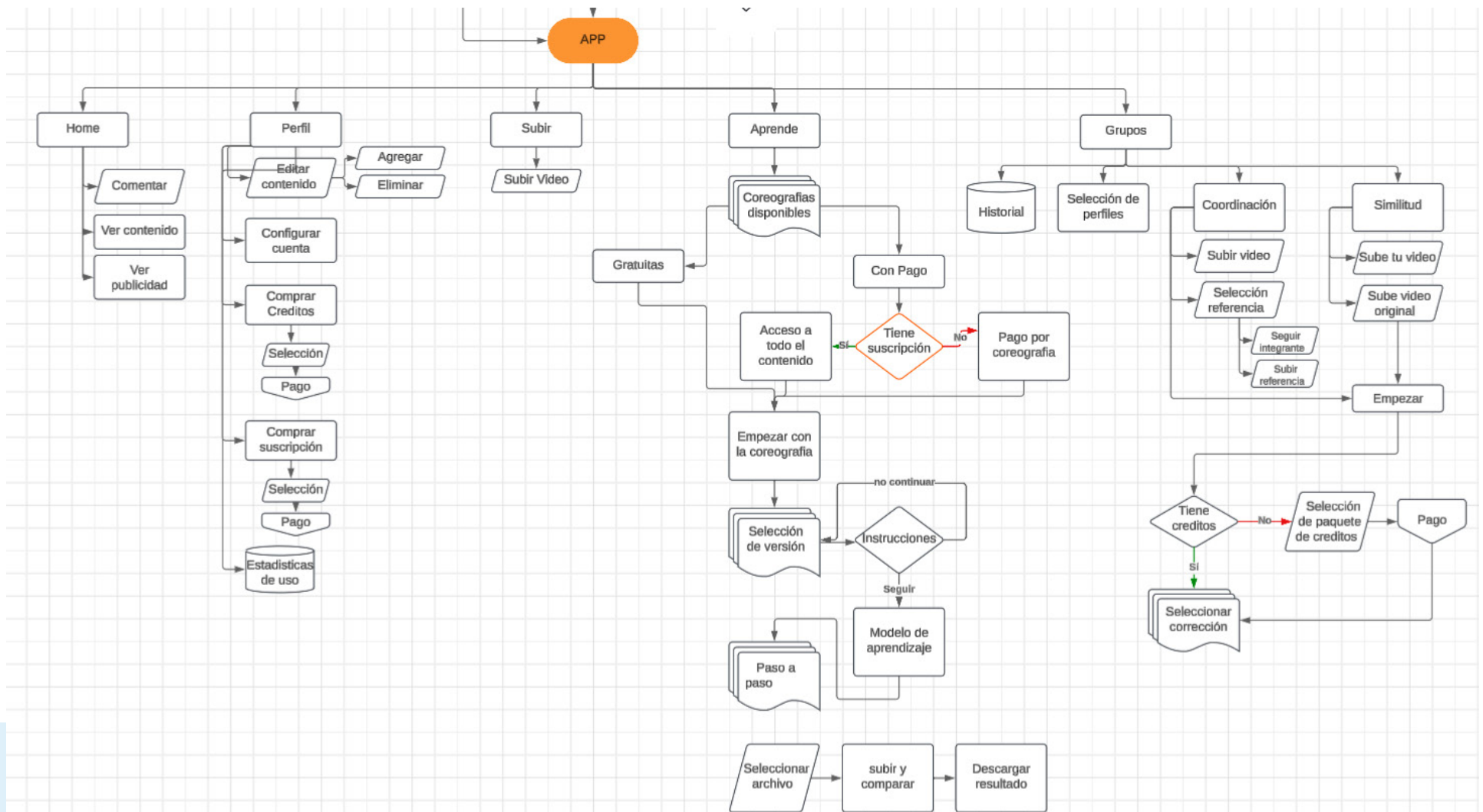


Imagen del Diagrama de flujo

El sistema interactivo propuesto para Dancistic se estructura en un flujo jerárquico que abarca desde el acceso inicial del usuario hasta las funciones avanzadas de aprendizaje, interacción y evaluación. Este diseño busca garantizar una navegación intuitiva, optimizando la experiencia de uso y alineándose con los objetivos generales y específicos del proyecto.

El recorrido comienza en la pantalla de inicio, donde el usuario puede iniciar sesión con credenciales propias o mediante plataformas externas como Google o Facebook, o bien optar por crear una cuenta. Durante este proceso, se ofrecen alternativas como la suscripción directa, la selección de un plan o el acceso a una prueba gratuita. El sistema gestiona los procesos de pago de manera integrada y segura.

Una vez autenticado, el usuario accede a la pantalla principal (Home), que se ramifica en cuatro áreas principales:

- Home: Permite comentar, ver contenido y acceder a publicidad.
- Perfil: Contiene herramientas para editar contenido, configurar la cuenta, comprar o modificar suscripciones, acceder a estadísticas de uso y administrar configuraciones de ayuda, permisos, privacidad, accesibilidad e idioma.
- Aprende: Presenta un listado de coreografías disponibles. El acceso a versiones completas requiere suscripción o pago individual, mientras que la versión general es de libre acceso. Incluye opciones para iniciar el aprendizaje, evaluar la coreografía y obtener estadísticas de rendimiento, con posibilidad de repetir o continuar.
- Grupos: Facilita la selección de perfiles y la interacción en modos de coordinación o similitud, permitiendo subir vídeos de

referencia y originales, así como recibir correcciones mediante un sistema de créditos.

- Subir contenido: Permite subir archivos de vídeo.

Este sistema interactivo integra la lógica del árbol estructural y el diagrama de flujo en un mismo modelo, asegurando que cada decisión y acción del usuario esté contemplada. Las rutas contemplan tanto las funciones principales como las alternativas en caso de no contar con suscripción o créditos, garantizando así una experiencia inclusiva y adaptativa.

La estructuración de este sistema responde directamente a los objetivos del proyecto:

Facilitar el acceso a contenido de aprendizaje interactivo.
Proporcionar herramientas de evaluación y mejora continua.
Fomentar la interacción social y la colaboración entre usuarios.
Mantener una interfaz clara y accesible que reduzca la curva de aprendizaje.

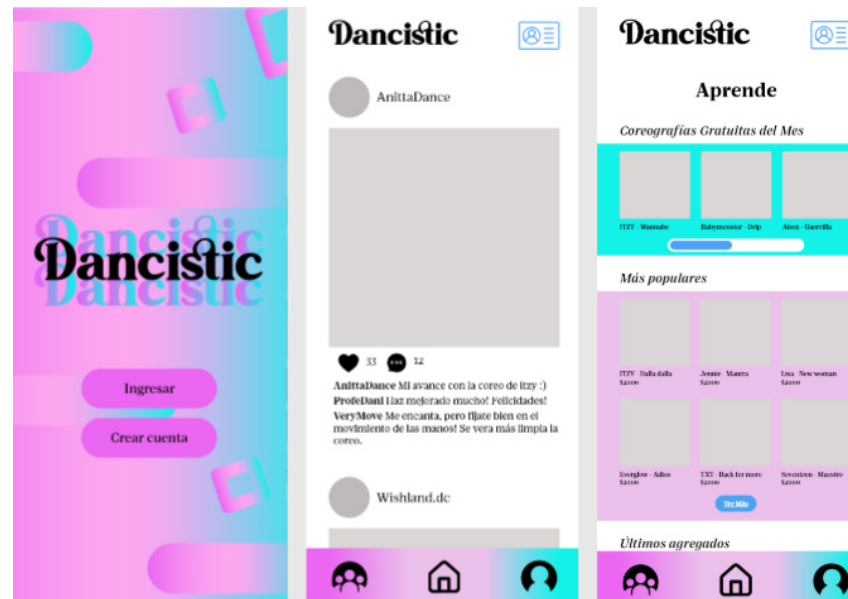
Tecnologías y Especificaciones Clave

Capa	Tecnología	Justificación
Frontend	HTML5, CSS3 (Flexbox + Grid), Vanilla JS	Ligero y fácil de mantener.
Backend	Flask (Python 3.11)	Sencillo para prototipado rápido y fácil integración con MediaPipe.
Análisis de movimiento	MediaPipe Pose + NumPy + SciPy	Eficiencia en CPU, sin necesidad de GPU dedicada.
Procesamiento de vídeo	OpenCV + FFmpeg (libx264)	Aceleración por hardware, compatibilidad multiplataforma.
Persistencia	SQLite en prototipo; escalable a PostgreSQL	Suficiente para MVP, migrable a producción.

Iteraciones y Validaciones

- Prototipo 1: Vídeo de visibilidad y validación del flujo de aprendizaje actual, mostrando al final la solución propuesta.
- Prototipo 2: Presentación del diseño inicial de la plataforma.
- Prototipo 3: Rediseño de la plataforma según requerimientos y comentarios entregados.
- Prototipo 4: MVP con MediaPipe, test con 10 bailarines, obteniendo un 85% de satisfacción en claridad de feedback.
- Prototipo 5: MVP plataforma completa con MediaPipe y Lovable.

Imagenes Primeros prototipos



Comparador de Poses

1 Video de Referencia

En esta parte puedes ver el video base, con parte de la coreografía que te quieres aprender. Puedes verlo las veces que quieras



2 Sube tu intento

Para esta parte necesitaras grabarte haciendo esa parte de la coreografía y subirla aqui abajo, para asi poder hacer la comparación entre tu video y el de Referencia Se te dara un porcentaje de similitud si es mayor a 80, podras avanzar con la siguiente parte. Puedes descargar el resultado si asi gustas.

Seleccionar archivo Ningún archivo seleccionado Subir y Comparar



Proyección y Escalabilidad

- Ampliación de estilos de baile disponibles.
- Optimización del aprendizaje para hacerlo más cercano a experiencias como Dance Central, con retroalimentación en tiempo real.
- Alianzas con profesores o academias para integrar coreografías dentro de la app.
- Adaptación a otros entornos distintos al baile, como deportes.

Modelo de Negocios

El modelo de negocios de Dancistic se fundamenta en un enfoque mixto que combina acceso gratuito con funcionalidades limitadas (freemium) y un sistema de monetización basado en suscripciones, microtransacciones y un marketplace de contenido digital.

Fuentes de ingresos:

- Modelo freemium con funciones básicas y análisis limitados a cambio de ver anuncios.
- Compra de créditos para análisis adicionales (precio estimado: \$500 CLP por crédito).
- Suscripción mensual premium (\$25.000 CLP) con acceso ilimitado a funciones avanzadas.
- Marketplace de coreografías con compra individual.

Costos principales:

- Desarrollo y mantenimiento de la plataforma.
- Infraestructura en la nube.
- Marketing digital.
- Soporte y alianzas con creadores.

Simulación de financiamiento FONDART

Subsidio: \$30.000.000 CLP inicial + aporte propio de \$6.000.000 CLP.

Asignación:

- \$18.000.000 CLP: Desarrollo tecnológico y prototipo funcional.
- \$6.000.000 CLP: Producción audiovisual de coreografías.
- \$3.000.000 CLP: Gestión administrativa y gastos operativos.
- Aporte propio: Marketing digital inicial y formalización legal.

Supuestos de crecimiento:

MVP lanzado al mes 10.

Crecimiento de usuarios freemium: 3.000 en Año 0, 150.000 en Año 5.

Conversión a suscripción: 4 % en Año 1, 25 % en Año 5.

Marketplace a partir del Año 2.

Proyección financiera (resumen):

Año 1: Ingresos \$55M – Costos \$24M - Flujo neto \$31M

Año 2: Ingresos \$399M – Costos \$31.2M - Flujo neto \$367.8M

Año 3: Ingresos \$1.53B – Costos \$40.6M - Flujo neto \$1.489B

Año 4: Ingresos \$3.175B – Costos \$71M - Flujo neto \$3.104B

Año 5: Ingresos \$5.685B – Costos \$100M - Flujo neto \$5.585B

Conclusión:

El financiamiento FONDART permite a Dancistic iniciar operaciones sin endeudamiento ni pérdida de participación accionaria, asegurando el desarrollo e introducción al mercado. La proyección a cinco años demuestra viabilidad económica y potencial de escalamiento, manteniendo un impacto cultural y comunitario alineado con el subsidio.

Definición Línea Gráfica Interfaz

La interfaz de Dancistic se diseña bajo un enfoque minimalista y limpio, priorizando la claridad en la interacción y la jerarquía visual. La paleta de colores definida se basa en tonos suaves y armónicos de rosado, lila, celeste y blanco, evitando el uso de amarillos o verdes para mantener coherencia estética y transmitir una sensación de modernidad y frescura. Estos colores se aplican en forma de gradientes sutiles, que aportan dinamismo y profundidad sin saturar la pantalla.

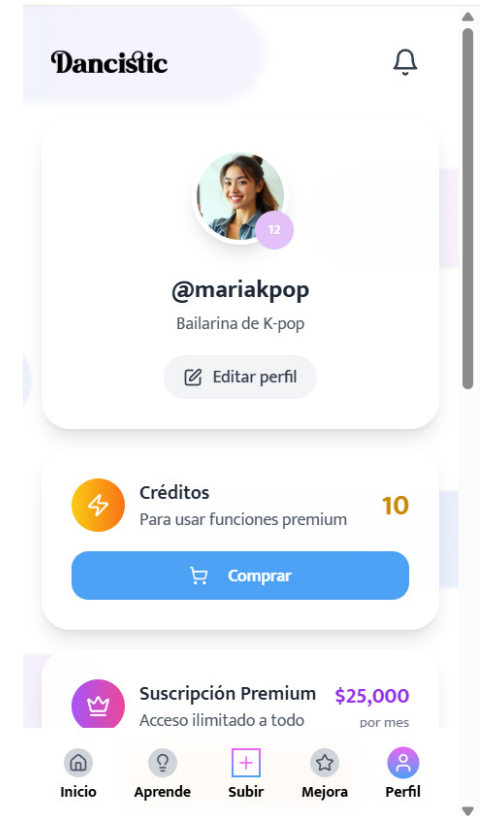
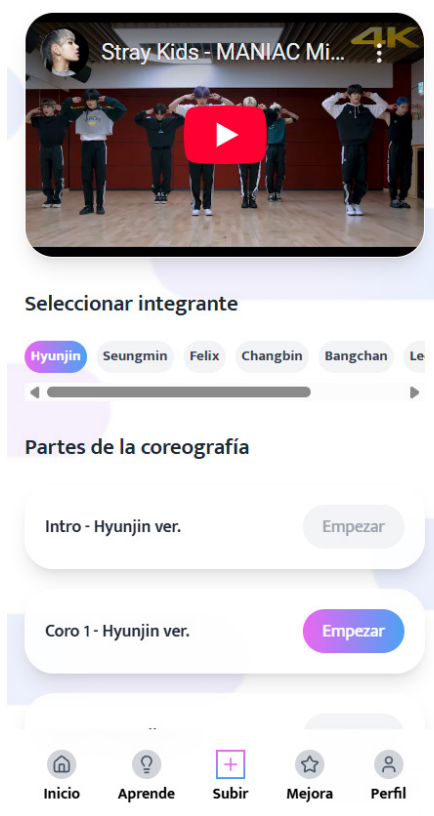
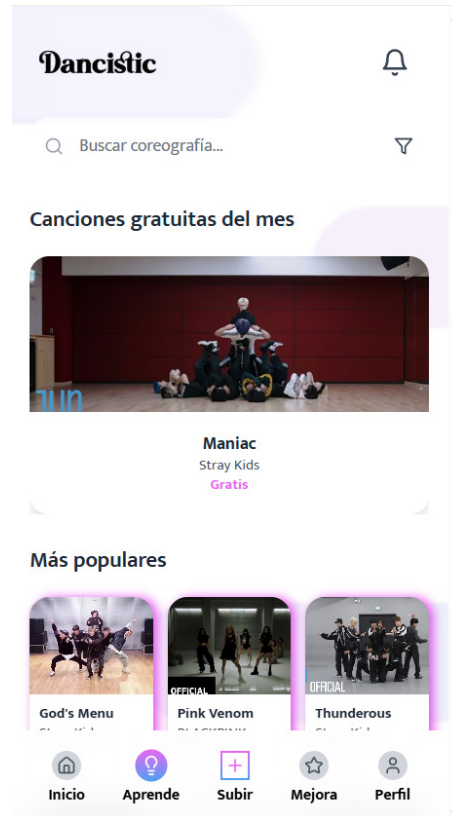
La tipografía principal, utilizada en el logotipo, es de estilo elegante y distintivo, lo que aporta identidad de marca. En los textos funcionales y botones se utilizan tipografías sans serif limpias, optimizadas para legibilidad en entornos digitales.

Los botones primarios destacan con rellenos en gradiente rosado-lila y bordes redondeados, generando un aspecto amigable y accesible. Los botones secundarios y elementos interactivos mantienen un diseño simple, con bordes finos y relleno blanco o semitransparente, asegurando que el foco visual se mantenga en las acciones clave.

El uso de iconografía minimalista en la barra de navegación inferior (Inicio, Aprende, Subir, Mejora, Perfil) facilita la identificación de secciones y mejora la experiencia de usuario. Las tarjetas de contenido presentan esquinas redondeadas, sombras suaves y fondos blancos, lo que permite que las imágenes y vídeos destaquen como elemento principal.

Además, se emplea una jerarquía clara de espaciado y tamaños tipográficos, que guía al usuario desde los elementos más importantes (títulos, botones de acción) hasta los secundarios (descripciones, etiquetas). La disposición de los elementos favorece la navegación intuitiva, manteniendo consistencia entre pantallas como inicio de sesión, feed social, catálogo de coreografías y reproductor de clases.

En conjunto, la línea gráfica de Dancistic combina armonía cromática, simplicidad visual y acentos gradientes para ofrecer una experiencia atractiva, moderna y coherente con los valores del proyecto: accesibilidad, creatividad y comunidad.



Imagenes Prototipo Dancistic

Implementación

El sistema interactivo de Dancistic se implementa como una aplicación web desarrollada con Flask (Python) para el backend, HTML/CSS/JavaScript para el frontend y MediaPipe para la detección de poses. Admite videos en formato .mp4 o .mov, con duración máxima de 3 minutos, y funciona tanto en desktop como en dispositivos móviles.

Protocolo de Uso

1. Registro o inicio de sesión: El usuario crea una cuenta o accede con credenciales existentes, pudiendo iniciar sesión también con Google o Facebook.
2. Selección de coreografía: Desde la sección Aprende, el usuario elige la canción y coreografía de interés, filtrando por integrante si corresponde.
3. Visualización guiada: La app muestra el video de referencia junto con las partes desglosadas de la coreografía (intro, coro, versos).
4. Grabación o carga del intento: El usuario sube su video para análisis.
5. Procesamiento y comparación: El sistema analiza el video, calcula porcentajes de similitud por parte del cuerpo y genera un reporte narrativo con sugerencias de mejora.
6. Revisión y seguimiento: El usuario recibe el resultado y puede repetir el proceso para mejorar su desempeño.

A continuación se describe el proceso completo de desarrollo del modelo de análisis de similitud de coreografías utilizado en el proyecto Dancistic. Desde los primeros experimentos con OpenPose hasta la implementación final con MediaPipe, se detalla cada decisión técnica, los desafíos enfrentados, las iteraciones realizadas y los cambios en los objetivos iniciales.

Primeros experimentos con OpenPose

El desarrollo comenzó utilizando OpenPose, un sistema de código abierto para la detección de poses humanas a partir de videos. Se eligió inicialmente por su precisión en la identificación de puntos del cuerpo. En esta etapa se utilizó un script en Python que procesaba dos videos: uno de referencia (la coreografía original) y otro del usuario intentando imitarla.

El código extraía las coordenadas de los puntos clave del cuerpo (keypoints) y luego se normalizaban utilizando la distancia entre los hombros. Se implementó Dynamic Time Warping (DTW) para comparar secuencias de poses. Sin embargo, OpenPose presentaba problemas críticos:

- Tiempos de procesamiento muy altos incluso para videos cortos.
- Requisitos de hardware (especialmente GPU) que limitaban su portabilidad.
- Dificultad para integrarlo de forma eficiente en una aplicación web.

Esto motivó un cambio de enfoque para probar alternativas más ligeras y rápidas.

Transición a MediaPipe

Se decidió adoptar MediaPipe de Google como alternativa a OpenPose. Esta librería ofrece un modelo de pose mucho más eficiente, fácil de instalar, y que permite trabajar en tiempo real con buena precisión.

El primer paso fue replicar el procesamiento de vídeo usando MediaPipe. Esto incluyó:

- Extracción de 33 puntos clave por cuadro.
- Normalización de coordenadas tomando como centro el punto medio entre los hombros.
- Almacenamiento de las coordenadas para el vídeo de referencia en un archivo .npy.
- Aplicación de DTW para comparar el intento del usuario con la referencia.

Este cambio permitió acelerar el procesamiento drásticamente y facilitó la construcción de una aplicación web con Flask.

Dificultades encontradas

Durante el desarrollo con MediaPipe surgieron varios desafíos importantes:

- Sincronización de la cantidad de frames entre vídeos de diferente duración.
- Desalineación visual entre esqueletos cuando se intentó dibujar el esqueleto de referencia encima del usuario.
- Diferencias en proporciones físicas entre usuarios que dificultaban una comparación visual justa.
- Discrepancias en la velocidad de reproducción entre los vídeos procesados y el original.
- Variaciones en el número de frames detectados con landmarks válidos.

Inicialmente se intentó resolver esto superponiendo el esqueleto azul (de referencia) al esqueleto del usuario en cada cuadro. Para ello se aplicaron técnicas de escalado, rotación y alineación usando los hombros como puntos de anclaje. Sin embargo, los resultados fueron inconsistentes: el esqueleto azul se deformaba y no lograba coincidir bien con el del usuario, dificultando la

interpretación visual.

Cambio en el enfoque de análisis

Dado que la superposición visual resultaba poco confiable, se decidió cambiar el enfoque. En lugar de mostrar visualmente los esqueletos, se optó por generar un análisis narrativo por partes del cuerpo.

El cuerpo se dividió en cinco grupos:

- Brazos
- Piernas
- Cabeza
- Tronco
- General

Para cada grupo se calculó la distancia promedio entre los puntos clave del usuario y la referencia a lo largo del tiempo. Con esta información se generó un reporte textual que incluye:

- Nivel de similitud por parte del cuerpo.
- Segundos en los que se detectaron mayores desviaciones.
- Comentarios que indican si se debe mejorar o si el rendimiento fue bueno.

Resultados

El sistema final permite:

- Subir un vídeo de intento.
- Comparar automáticamente con una referencia.
- Obtener un porcentaje general de similitud.
- Recibir un análisis narrativo por partes del cuerpo con sugerencias.

Esta solución demostró ser más eficaz y clara para el usuario final, al evitar problemas visuales y centrarse en un análisis por texto fácil de entender.

Resultados e Impacto Esperados

Se espera que el sistema interactivo de Dancistic tenga un impacto significativo en la forma en que los usuarios aprenden, practican y perfeccionan coreografías.

En primer lugar, el análisis automatizado por partes del cuerpo permitirá que cada usuario identifique con claridad cuáles son sus áreas de mejora, evitando interpretaciones ambiguas y ofreciendo un feedback específico y accionable.

Esta retroalimentación personalizada no solo optimiza el proceso de aprendizaje, sino que también fomenta la autoconfianza y la motivación para continuar practicando.

Además, la plataforma contribuirá a democratizar el acceso a herramientas avanzadas de entrenamiento, ya que podrá utilizarse desde cualquier dispositivo con conexión a internet, sin requerir equipamiento especializado ni conocimientos técnicos previos. Esto abre la posibilidad de que tanto bailarines aficionados como profesionales puedan beneficiarse de la solución, así como instructores que deseen incorporar un sistema de evaluación objetiva a sus clases.

En un contexto más amplio, se proyecta que la adopción de Dancistic incremente la participación en comunidades de baile en línea, fortaleciendo el sentido de pertenencia y la colaboración entre usuarios de distintos lugares. La herramienta también podría convertirse en un recurso valioso para academias de danza y eventos competitivos, al proporcionar una medición estanda-

rizada del desempeño.

En resumen, el sistema no solo facilitará el aprendizaje y perfeccionamiento técnico de las coreografías, sino que también tendrá un impacto positivo en la accesibilidad, la inclusión y la conexión entre personas apasionadas por el baile, generando así un aporte tangible tanto a nivel individual como comunitario.

Observaciones de Testeos

Observaciones de testeos

Durante las pruebas con usuarios, se identificaron mejoras necesarias tanto en la presentación de la coreografía como en el sistema de retroalimentación. Los participantes sugirieron incorporar más detalles sobre la ejecución de cada paso, priorizando explicaciones claras por sobre simples acercamientos a la parte superior o inferior del cuerpo. También se propuso la implementación de un sistema de repetición en bucle que inicie de forma lenta y aumente progresivamente la velocidad, permitiendo un aprendizaje más gradual. En cuanto al análisis de resultados, se recomendó representar gráficamente las métricas de similitud e incluir enlaces que lleven directamente al segundo exacto donde se produjo un error, para facilitar su revisión. Asimismo, se destacó la utilidad de añadir, al inicio de la sección de aprendizaje y comparación, un mensaje o tutorial explicativo sobre el funcionamiento de la herramienta. Entre las mediciones obtenidas, la cabeza presentó un 87,39% de similitud sin errores destacados, los brazos un 76,1% con errores recurrentes en momentos específicos, el torso un 94,7% sin errores, y las piernas un 90,71% con algunas desviaciones puntuales. Estos resultados permiten

identificar con precisión las áreas de mayor dificultad y orientar el ajuste del nivel de exigencia para optimizar la experiencia de aprendizaje.

Re-definiciones

A partir de las observaciones recogidas durante las pruebas con usuarios, se realizaron ajustes clave en el sistema para responder de manera efectiva a las necesidades detectadas. En primer lugar, se cambió el sistema de aprendizaje que se tenía, en vez de mostrar el vídeo separado por manos y pies, se decidió usar un vídeo tutorial encontrado en youtube, si bien depender de Youtube no es el objetivo de Dancistic, fue una forma rápida y eficaz de solucionar este apartado, lo que facilita el aprendizaje y la asimilación de cada paso de la coreografía. Además, se reemplazó el formato exclusivamente textual del análisis por un sistema gráfico que permite visualizar de manera más clara las métricas de similitud y las desviaciones detectadas, brindando al usuario una retroalimentación más intuitiva. Estas mejoras cumplen con los requerimientos solicitados durante el testeó y refuerzan la experiencia interactiva, haciendo que el proceso de aprendizaje sea más accesible, visual y preciso.

Definición de Parámetros

El proyecto Dancistic se evaluó en base a cuatro ejes fundamentales: factibilidad técnica, viabilidad económica, gestión de negocio y posicionamiento en el mercado.

Factibilidad técnica: El sistema ha demostrado un funcionamiento estable mediante tecnologías accesibles y eficientes como MediaPipe y Flask, que permiten el análisis de similitud en tiempo real sin necesidad de equipos de alto rendimiento. La implementación de reproducción en bucle y visualización gráfica de resultados confirma la capacidad técnica para integrar mejoras sin comprometer la experiencia de usuario.

Viabilidad económica: El desarrollo de la plataforma ha requerido recursos moderados en infraestructura, ya que se basa en herramientas de código abierto y en un modelo escalable que puede funcionar inicialmente en servidores de costo medio. Esto permite implementar un modelo de monetización gradual, combinando acceso gratuito a funciones básicas con planes premium para funcionalidades avanzadas, como análisis detallados, contenido exclusivo y asistencia personalizada.

Gestión de negocio: La propuesta se alinea con un modelo de suscripción mensual accesible para el público objetivo, complementado con posibles ingresos por publicidad segmentada y colaboraciones con academias de baile. La arquitectura modular del sistema permite añadir nuevas funciones y expandir la oferta sin re-diseñar la plataforma desde cero, asegurando un crecimiento sostenible.

Posicionamiento en el mercado: Dancistic se sitúa como una solución innovadora en el nicho del aprendizaje de coreografías, combinando análisis automatizado, retroalimentación visual y experiencia gamificada. Su diferenciador principal es la capacidad de evaluar el desempeño del usuario de forma precisa y visual, ofreciendo una experiencia más inmersiva que las plataformas tradicionales de tutoriales en vídeo. Esta propuesta de valor permite competir tanto en el mercado local como inter-

nacional, con especial atractivo para comunidades de K-pop y academias de danza.

Proyecciones

El desarrollo futuro de Dancistic contempla una serie de mejoras estratégicas que permitirán ampliar su alcance, optimizar la experiencia de aprendizaje y fortalecer su posicionamiento en el mercado.

En el corto plazo, se buscará incorporar nuevos estilos de baile más allá del K-pop, incluyendo danzas urbanas, contemporáneas y latinas, con el objetivo de atraer a un público más diverso. Asimismo, se trabajará en la optimización del sistema de aprendizaje para que se asemeje a la experiencia de videojuegos como Dance Central, integrando retroalimentación en tiempo real que permita al usuario ajustar su ejecución mientras practica.

En el mediano plazo, se establecerán alianzas con profesores y academias de baile, de manera que puedan compartir sus propias coreografías dentro de la aplicación, generando contenido exclusivo y fomentando una comunidad activa de aprendizaje colaborativo.

En el largo plazo, Dancistic buscará consolidarse como una plataforma integral para la enseñanza y práctica de coreografías a nivel internacional, incorporando tecnologías de análisis en vivo más precisas, herramientas de gamificación y opciones de personalización avanzada para que cada usuario pueda adaptar su experiencia de aprendizaje a sus objetivos y nivel técnico. Además, se proyecta que el modelo de análisis desarrollado para Dancistic pueda ser adaptado a otros rubros, como el deporte, la rehabilitación física y la preparación física persona-

lizada. Esta expansión permitiría aplicar la misma tecnología de captura y comparación de movimiento para optimizar entrenamientos, apoyar procesos de recuperación y mejorar el rendimiento físico en distintos contextos profesionales y recreativos. Con el sistema interactivo ya definido y las funcionalidades principales implementadas, el siguiente paso fue documentar en detalle la estructura técnica que hace posible el análisis de similitud en Dancistic. A continuación, se presenta la explicación completa del código, desglosando cada uno de sus componentes y procesos internos. Este apartado detalla desde la configuración inicial del entorno y la carga de modelos, hasta la lógica de procesamiento de vídeos, la extracción y normalización de puntos clave, la comparación mediante Dynamic Time Warping y la generación de reportes de resultados. El objetivo es que cualquier lector, incluso sin un conocimiento técnico avanzado, pueda comprender cómo se integran las distintas tecnologías (Flask, OpenCV, MediaPipe, entre otras) para lograr el funcionamiento final de la plataforma.

Explicación del Código

Se procede a dar la explicación detallada del código del modelo de Dancistic, explicando los 3 archivos python que lo componen y la funcionalidad del mismo dentro de Dancistic, preprocess.py, main.py y app.p

preprocess.py — Explicación Completa

Este script no se ejecuta junto al servidor Flask, sino una sola vez antes del uso del Sistema para crear el archivo de referencia (reference_coords.npy) que será utilizado para comparar movimientos en app.py.

- **Importaciones**

```
# preprocess_reference.py
import cv2
import mediapipe as mp
import numpy as np
```

- cv2: biblioteca de visión computacional para cargar y procesar el vídeo.
- mediapipe: modelo de Google para extraer puntos clave de la pose humana.
- numpy: para almacenar y guardar los puntos clave en formato .npy.

- **Inicialización del modelo de pose**

```
mp_pose = mp.solutions.pose
pose = mp_pose.Pose()
```

- Se activa la solución de pose de MediaPipe.
- Aquí no se configura `min_detection_confidence` ni `static_image_mode`, por lo que toma los valores por defecto: el modelo buscará poses en vídeo en modo dinámico (ideal para vídeo continuo).
- **Lista para guardar las coordenadas**

```
coords_list = []
```

Lista vacía donde se irán agregando los 33 landmarks detectados por frame, con sus coordenadas (x, y, z) y visibility.

- **Apertura del vídeo de referencia y lectura frame por frame**

```
cap = cv2.VideoCapture('static/hyunjin.mp4')
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = pose.process(frame_rgb)
    if results.pose_landmarks:
        landmarks = results.pose_landmarks.landmark
        frame_coords = np.array([[lm.x, lm.y, lm.z, lm.visibility] for lm in landmarks])
        coords_list.append(frame_coords)
```

- Carga el vídeo base que contiene la coreografía de referencia que será imitada por los usuarios.
- Se leen los frames uno por uno.
- Cuando ya no quedan más (ret == False), se detiene el bucle.
- Convierte el frame a RGB porque MediaPipe espera imágenes en ese formato.
- Procesa el frame y guarda los resultados (landmarks detectados).
- Si se detectan landmarks:
 - Se extraen los valores (x, y, z, visibility) de cada uno de los 33
 - puntos.
- Se convierte en un array y se guarda en la lista coords_list
- **Finalización del vídeo y guardado de resultados**

```
cap.release()

coords_array = np.array(coords_list)
np.save('static/reference_coords.npy', coords_array)
print("Coordenadas del video de referencia guardadas correctamente.")
```

- Libera el vídeo para liberar memoria.
- Convierte la lista en un array 3D de NumPy con forma (n_frames, 33, 4).
- Se guarda en un archivo .npy llamado reference_coords.npy, que luego será
- cargado automáticamente por app.py cuando arranque el servidor Flask.
- Muestra un mensaje de confirmación.
- Si cambias el vídeo de referencia (una nueva coreografía), debes volver a ejecutar este script para generar un nuevo archivo reference_coords.npy.
- El archivo .npy generado es luego cargado automáticamente por app.py para comparar los vídeos de los usuarios.

¿Qué función cumple preprocess.py en el sistema?

Es el preprocesador de referencia, necesario una sola vez por cada coreografía. Esto significa:

main.py — Explicación Completa

- **Importaciones**

```
import tkinter as tk
from tkinter import filedialog, messagebox
import cv2
import mediapipe as mp
import numpy as np
import time
import os
from threading import Thread
```

- tkinter: módulo estándar de Python para crear interfaces gráficas.
- filedialog: permite al usuario seleccionar archivos desde el sistema.
- messagebox: muestra mensajes emergentes (errores, alertas, etc).
- cv2: OpenCV, usado para lectura y reproducción de vídeo.
- mediapipe: para detección de la pose humana.
- numpy: manejo de datos numéricos y arrays.
- time: pausas o control temporal.
- os: operaciones del sistema.
- threading: se usa para que el vídeo se reproduzca sin congelar la interfaz (ejecución en segundo plano).

- Inicialización de MediaPipe Pose y variables de referencia y control

```
# Inicializa MediaPipe Pose
mp_pose = mp.solutions.pose
pose = mp_pose.Pose(static_image_mode=False, min_detection_confidence=0.5, min_tracking_confidence=0.5)
mp_drawing = mp.solutions.drawing_utils

# Inicializa coordenadas de referencia
reference_landmarks = None

# Variables de control
video_reference_path = ""
video_playback_speeds = [1.0, 0.75, 0.5]
correct_attempts = 0
```

- El código comienza configurando MediaPipe Pose, una librería que permite detectar puntos clave del cuerpo en imágenes y vídeos. Se crea un objeto llamado pose que se encargará de reconocer la posición del cuerpo en cada fotograma, con parámetros de confianza mínima tanto para la detección inicial como para el seguimiento de los movimientos. Además, se carga un módulo auxiliar (mp_drawing) que sirve para dibujar en pantalla las poses detectadas.
- Se definen varias variables que permiten manejar el funcionamiento del sistema:

- reference_landmarks se usa para almacenar las coordenadas del cuerpo del vídeo de referencia.

- video_reference_path guarda la ruta del archivo de vídeo

que el usuario seleccione.

- video_playback_speeds contiene una lista de velocidades en que el vídeo podrá reproducirse (normal, 0.75 y 0.5).

- correct_attempts contabiliza los intentos correctos del usuario.

- Extracción de coordenadas corporales y calculo de similitud

```
# Función para extraer coordenadas de un frame
def extract_landmarks(image):
    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    result = pose.process(image_rgb)
    if result.pose_landmarks:
        return [(lm.x, lm.y, lm.z) for lm in result.pose_landmarks.landmark]
    return None

# Calcular similitud (por simplicidad, distancia euclidiana)
def calculate_similarity(landmarks1, landmarks2, tolerance=0.1):
    if landmarks1 is None or landmarks2 is None:
        return 0.0
    landmarks1 = np.array(landmarks1)
    landmarks2 = np.array(landmarks2)
    diffs = np.linalg.norm(landmarks1 - landmarks2, axis=1)
    similar_points = np.sum(diffs < tolerance)
    return similar_points / len(landmarks1)
```

- La función `extract_landmarks` recibe una imagen y la convierte a formato RGB (porque OpenCV trabaja en BGR y MediaPipe en RGB). Luego, con `pose.process()`, se analizan los puntos clave del cuerpo. Si se encuentran poses, devuelve una lista con las coordenadas tridimensionales (x, y, z) de cada punto; en caso contrario, devuelve `None`.
- La función `calculate_similarity` compara dos poses (listas de coordenadas) y mide cuánto se parecen. Para esto, usa la dis-

tancia euclidiana entre cada punto del cuerpo de ambas poses. Si la distancia entre dos puntos es menor que un valor de tolerancia, se considera que esos puntos son similares. Finalmente, devuelve un valor entre 0 y 1 que representa el porcentaje de puntos coincidentes entre ambas poses.

- Reproducción de vídeo de referencia

```
# Reproducir video de referencia a diferentes velocidades
def play_reference_video():
    global video_reference_path, reference_landmarks

    if not video_reference_path:
        messagebox.showerror("Error", "Primero selecciona un video de referencia.")
        return

    cap = cv2.VideoCapture(video_reference_path)
    success, frame = cap.read()
    if not success:
        messagebox.showerror("Error", "No se pudo abrir el video.")
        return

    reference_landmarks = extract_landmarks(frame)

    for speed in video_playback_speeds:
        cap = cv2.VideoCapture(video_reference_path)
        while cap.isOpened():
            ret, frame = cap
```

La función `play_reference_video` es la encargada de mostrar el vídeo que el usuario elija. Primero revisa si hay un archivo seleccionado; si no, muestra un mensaje de error. Luego abre el vídeo con OpenCV (`cv2.VideoCapture`) y extrae las coordenadas de la primera imagen para usarlas como referencia.

Después, reproduce el mismo vídeo en distintas velocidades (normal, lenta y más lenta). De esta forma, el sistema permite que el

usuario observe la coreografía en diferentes tiempos, lo que facilita el aprendizaje y la comparación de movimientos.

Este código establece la base del sistema de análisis de danza: selecciona un vídeo de referencia, detecta los puntos clave del cuerpo con MediaPipe, compara movimientos mediante cálculos matemáticos y ofrece la posibilidad de ver la coreografía a distintas velocidades para un aprendizaje más guiado.

app.py — Explicación Completa

El archivo app.py es la columna vertebral del sistema web de análisis de poses.

Funciona como un servidor Flask que:

1. Recibe un vídeo de intento de un usuario a través de una interfaz web.
2. Procesa dicho vídeo usando MediaPipe Pose para obtener

coordenadas de puntos

3. corporales.
4. Compara esos puntos con un vídeo de referencia ya procesado.
5. Calcula una similitud por partes del cuerpo.
6. Genera un vídeo con los errores destacados.
7. Devuelve el resultado como un JSON al frontend con la similitud, el vídeo generado y un análisis textual.

• Importaciones

```
from flask import Flask, render_template, request, jsonify
import cv2
import numpy as np
import mediapipe as mp
import os
import subprocess
import json
from scipy.interpolate import interp1d
from werkzeug.utils import secure_filename
```

- `from flask import Flask, render_template, request, jsonify` - Sirve para crear la aplicación web, renderizar páginas HTML, recibir datos del usuario y devolver respuestas en formato JSON.
 - `import cv2` - Librería OpenCV, usada para procesar imágenes y videos.
 - `import numpy as np` - Para cálculos numéricos y manejo de arreglos de datos (coordenadas del cuerpo).
 - `import mediapipe as mp` - Para la detección y seguimiento de poses del cuerpo humano.
 - `import os` - Manejo de archivos y rutas dentro del sistema operativo.
 - `import subprocess` - Permite ejecutar procesos o comandos externos desde el programa.
 - `import json` - Para trabajar con datos estructurados en formato JSON (guardar/cargar información).
 - `from scipy.interpolate import interp1d` - Para interpolación matemática, útil para ajustar datos de movimiento.
 - `from werkzeug.utils import secure_filename` - Para asegurar que los archivos subidos por los usuarios tengan nombres válidos y seguros.
- **Carpetas y parametros iniciales del sistema**

```

app = Flask(__name__)

UPLOAD_FOLDER = 'uploads'
OUTPUT_FOLDER = 'static/outputs'
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
os.makedirs(OUTPUT_FOLDER, exist_ok=True)

reference_coords = np.load('static/reference_coords.npy', allow_pickle=True)

mp_pose = mp.solutions.pose
pose = mp_pose.Pose(static_image_mode=False, min_detection_confidence=0.5, min_tracking_confidence=0.5)

BODY_PARTS = {
    'cabeza': [0],
    'brazos': [11, 12, 13, 14, 15, 16],
    'torso': [23, 24],
    'piernas': [25, 26, 27, 28, 29, 30, 31, 32]
}
    
```

En esta parte primero, se definen los directorios UPLOAD_FOLDER y OUTPUT_FOLDER donde se almacenan respectivamente los vídeos cargados por el usuario y los resultados procesados, asegurando su creación automática con os.makedirs. Luego, se cargan desde un archivo .npy las coordenadas de referencia de una coreografía, que sirven como base de comparación para evaluar la ejecución del usuario. A continuación, se inicializa

MediaPipe Pose, estableciendo parámetros de confianza mínima tanto para la detección como para el seguimiento de los puntos corporales. Finalmente, se construye un diccionario denominado BODY_PARTS que organiza los índices de los landmarks de MediaPipe según regiones específicas del cuerpo (cabeza, brazos, torso y piernas), lo que facilita la segmentación y el análisis detallado del movimiento por cada parte.

- **Medir similitud y normalizar coordenadas corporales**

```
def penalized_similarity(diffs):
    penalized = []
    for d in diffs:
        if d < 0.07:
            penalized.append(0.0)
        elif d < 0.10:
            penalized.append(0.6)
        elif d < 0.20:
            penalized.append(0.85)
        else:
            penalized.append(1.0)
    avg_penalty = np.mean(penalized)
    similarity = max(0, 1 - avg_penalty**1.4)
    return similarity

def normalize_landmarks(landmarks):
    if landmarks.shape[0] == 0:
        return None
    origin = (landmarks[11, :3] + landmarks[12, :3]) / 2
    landmarks_norm = landmarks[:, :3] - origin
    height = np.linalg.norm(landmarks[0, :3] - origin)
    if height > 0:
        landmarks_norm /= height
    return landmarks_norm.flatten()
```

Aquí se implementan dos funciones clave para el análisis del movimiento. La primera, `penalized_similarity(diffs)`, calcula un índice de similitud entre dos secuencias de poses aplicando penalizaciones progresivas según el nivel de diferencia (`diffs`). Si la diferencia es muy baja, se considera como coincidencia casi perfecta (penalización mínima), mientras que diferencias mayores reciben penalizaciones crecientes hasta llegar a 1.0. El promedio de esas penalizaciones se usa para estimar la similitud final, ajustada con una potencia para dar más sensibilidad a los errores.

La segunda función, `normalize_landmarks(landmarks)`, se encar-

ga de normalizar las coordenadas de los puntos clave del cuerpo (`landmarks`). Primero establece un origen tomando como referencia los hombros (puntos 11 y 12), y luego ajusta todas las coordenadas restando ese origen. Después, calcula la altura del cuerpo usando la distancia entre la cabeza y el origen, y divide todos los puntos por esa altura para que la escala sea uniforme sin importar el tamaño o la distancia de la persona frente a la cámara. Finalmente, devuelve las coordenadas como un vector aplanado, listo para su comparación.

- **Detectar errores en los segundos de un vídeo**

```
def agrupar_errores(segundos, umbral=0.5):
    if not segundos:
        return []
    segundos_ordenados = sorted(segundos)
    errores_unicos = [segundos_ordenados[0]]
    for s in segundos_ordenados[1:]:
        if s - errores_unicos[-1] > umbral:
            errores_unicos.append(s)
    return errores_unicos

def segundos_a_timestamp(seg):
    minutos = int(seg // 60)
    segundos = int(seg % 60)
    return f"{minutos}:{segundos:02d}"
```

Este bloque define dos funciones que trabajan con los errores detectados en los segundos de un vídeo.

`agrupar_errores(segundos, umbral=0.5)`

Esta función recibe una lista de segundos donde ocurrieron errores y los organiza para evitar repeticiones demasiado cercanas. Primero, ordena la lista de segundos y comienza guardando el primer error. Luego, recorre los demás y solo agrega un nuevo segundo si la diferencia con el último error guardado es mayor al umbral (por defecto 0.5 segundos). De esta forma, si los errores aparecen muy juntos en el tiempo, se consideran como uno solo, lo que evita duplicados innecesarios en el análisis.

`segundos_a_timestamp(seg)`

Esta función convierte un tiempo en segundos a un formato de minutos y segundos legible. Calcula los minutos dividiendo los segundos por 60 y obtiene los segundos restantes con el módulo (%). Finalmente, devuelve el resultado en un formato de texto minutos:segundos, asegurando que los segundos siempre tengan dos dígitos (ejemplo: 1:05 en lugar de 1:5).

- **Rutas principales**

```
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/upload_attempt', methods=['POST'])
def upload_attempt():
    if 'video' not in request.files:
        return jsonify({'success': False, 'message': 'No se recibió ningún archivo de video.'}), 400

    file = request.files['video']
    if file.filename == '':
        return jsonify({'success': False, 'message': 'No se seleccionó ningún archivo.'}), 400

    filename = secure_filename(file.filename)
    input_path = os.path.join(UPLOAD_FOLDER, filename)
    file.save(input_path)

    output_filename = f'result_{filename}'
    output_path = os.path.join(OUTPUT_FOLDER, output_filename)
```

Este bloque define rutas principales de la aplicación Flask para manejar la interfaz y la subida de vídeos.

Ruta principal ('/')

La función `index()` se encarga de mostrar la página principal al usuario cuando accede a la aplicación. Lo hace renderizando el archivo `index.html`, que actúa como la interfaz gráfica inicial donde se puede interactuar con el sistema (por ejemplo, subir un vídeo).

Ruta de subida de vídeos ('/upload_attempt')

Esta función, `upload_attempt()`, se ejecuta cuando el usuario envía un archivo de vídeo mediante un formulario POST. Lo primero que hace es verificar si el archivo realmente fue envia-

do ('vídeo' not in `request.files`) y si el nombre no está vacío. Si hay algún problema, devuelve una respuesta en formato JSON indicando el error.

Si el archivo es válido, se asegura de que el nombre sea seguro usando `secure_filename` y luego guarda el archivo en la carpeta definida como `UPLOAD_FOLDER`. Posteriormente, genera un nombre de salida (`result_nombreArchivo`) y define la ruta en la carpeta `OUTPUT_FOLDER`, que servirá para guardar los resultados procesados.

- **Detectar errores en los segundos de un vídeo**

```
try:
    similarity, fixed_video_path, analysis_result_path, frame_data_path = process_and_compare(input_path, output_path)
except Exception as e:
    return jsonify({'success': False, 'message': f'Error durante la comparación: {str(e)}'}, 500)

with open(analysis_result_path, 'r') as f:
    analysis_text = f.read()

return jsonify({
    'success': True,
    'similarity': round(similarity * 100, 2),
    'video': '/' + fixed_video_path.replace('\\', '/'),
    'analysis': analysis_result_path,
    'frame_data': "/" + frame_data_path.replace('\\', '/'),
    'analysis_text': analysis_text,
    'message': '¡Bien hecho! Sigamos con la siguiente parte' if similarity >= 0.8 else 'Casi lo tienes, sigue intentando'
})
```

Este bloque maneja el proceso de comparación del vídeo subido con el de referencia y la devolución de resultados al usuario.

Ejecución del análisis (try / except)

La línea principal llama a la función `process_and_compare(input_path, output_path)`, la cual devuelve cuatro elementos:

- `similarity`: el porcentaje de similitud entre el vídeo del usuario y el de referencia.
- `fixed_video_path`: la ruta del vídeo procesado.
- `analysis_result_path`: el archivo con el reporte de análisis.
- `frame_data_path`: los datos específicos de cada cuadro (frames).

Si ocurre un error durante el análisis, se captura en el bloque `except` y se envía una respuesta JSON con un mensaje de error y el detalle de la excepción.

Lectura del análisis generado

Se abre el archivo `analysis_result_path` en modo lectura ('r') para cargar el contenido textual del análisis generado automáticamente, y se almacena en `analysis_text`.

Respuesta en formato JSON

Finalmente, se construye una respuesta estructurada en formato JSON con los siguientes datos:

- `success`: indica si el proceso fue exitoso.
- `similarity`: el porcentaje de similitud redondeado a dos decimales.
- `video`: la ruta del vídeo procesado, ajustada al formato de URL.
- `analysis`: ruta del archivo de resultados.
- `frame_data`: ruta con los datos por fotogramas.
- `analysis_text`: el texto leído del archivo de análisis.
- `message`: un mensaje motivacional adaptado a la similitud

obtenida:

- Si es mayor o igual a 0.8 (80%): “¡Bien hecho! Sigamos con la siguiente parte”.
- Si es menor a 0.8: “Casi lo tienes, sigue intentando”.

- Procesar y comparar

```
def process_and_compare(input_path, output_path):
    cap = cv2.VideoCapture(input_path)
    frames, attempt_coords = [], []
    A = None

    # Inicializamos estructura para registrar similitud frame a frame
    frame_similarities = {
        "cabeza": [],
        "torso": [],
        "brazos": [],
        "piernas": []
    }

    while True:
        ret, frame = cap.read()
        if not ret:
            break
        results = pose.process(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
        if results.pose_landmarks:
            lms = np.array([[lm.x, lm.y, lm.z, lm.visibility] for lm in results.pose_landmarks.landmark])
            norm = normalize_landmarks(lms)
            if norm is not None:
                attempt_coords.append(norm)
                frames.append((frame.copy(), lms))
                if A is None:
                    idx = [0,11,12,13,14,15,16,23,24,25,26,27,28]
                    ref_pts = np.array(reference_coords[0])[idx, :2]
                    usr_pts = lms[idx, :2]
                    h, w = frame.shape[:2]
                    A, _ = cv2.estimateAffinePartial2D(ref_pts*[w,h], usr_pts*[w,h], method=cv2.LMEDS)
```

Este bloque maneja el proceso de comparación del vídeo subido con el de referencia y la devolución de resultados al usuario.

Ejecución del análisis (try / except)

La línea principal llama a la función `process_and_compare(input_path, output_path)`, la cual devuelve cuatro elementos:

- `similarity`: el porcentaje de similitud entre el vídeo del usuario y el de referencia.
- `fixed_video_path`: la ruta del vídeo procesado.
- `analysis_result_path`: el archivo con el reporte de análisis.
- `frame_data_path`: los datos específicos de cada cuadro (frames).

Si ocurre un error durante el análisis, se captura en el bloque `except` y se envía una respuesta JSON con un mensaje de error y el detalle de la excepción.

Lectura del análisis generado

Se abre el archivo `analysis_result_path` en modo lectura ('r') para cargar el contenido textual del análisis generado automáticamente, y se almacena en `analysis_text`.

Respuesta en formato JSON

Finalmente, se construye una respuesta estructurada en formato JSON con los siguientes datos:

- `success`: indica si el proceso fue exitoso.
- `similarity`: el porcentaje de similitud redondeado a dos decimales.
- `video`: la ruta del vídeo procesado, ajustada al formato de URL.
- `analysis`: ruta del archivo de resultados.
- `frame_data`: ruta con los datos por fotogramas.
- `analysis_text`: el texto leído del archivo de análisis.
- `message`: un mensaje motivacional adaptado a la similitud

obtenida:

- Si es mayor o igual a 0.8 (80%): “¡Bien hecho! Sigamos con la siguiente parte”.
- Si es menor a 0.8: “Casi lo tienes, sigue intentando”.

- Procesar y comparar

```

fps = cap.get(cv2.CAP_PROP_FPS) or 30
cap.release()

if not frames:
    raise ValueError("No se detectaron landmarks en el video de intento.")
if A is None:
    raise ValueError("No se pudo calcular la transformación de referencia.")

if len(reference_coords) != len(frames):
    x_old = np.linspace(0, 1, len(reference_coords))
    x_new = np.linspace(0, 1, len(frames))
    aligned_reference_coords = interp1d(x_old, reference_coords, axis=0, kind='linear', fill_value="extrapolate")(x_new)
else:
    aligned_reference_coords = reference_coords

h, w = frames[0][0].shape[:2]
writer = cv2.VideoWriter(output_path, cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))
errores_por_parte = {p: [] for p in BODY_PARTS}
diffs_por_parte = {p: [] for p in BODY_PARTS}

```

En el fragment, se finaliza la preparación del vídeo y de los datos de referencia. Primero, se obtiene la tasa de fotogramas por segundo (fps) y se libera el objeto de captura con `cap.release()`. Luego, se aplican validaciones críticas: si no se detectaron landmarks en el vídeo o no se pudo calcular la transformación afín inicial (A), el proceso lanza un `ValueError`. Posteriormente, se verifica que la cantidad de frames procesados coincida con los del vídeo de referencia. En caso de diferencias, se ajusta la cantidad de coordenadas mediante interpolación lineal (inter-

`interp1d`), asegurando que ambos vídeos sean comparables cuadro a cuadro. Después, se obtienen las dimensiones del vídeo (h, w) y se inicializa un escritor de vídeo (`cv2.VideoWriter`) para generar el archivo de salida con la comparación visual. Finalmente, se crean estructuras vacías (`errores_por_parte` y `diffs_por_parte`) para registrar tanto los errores como las diferencias en cada segmento del cuerpo definido en `BODY_PARTS`.

En el fragmento, se recorre cada frame del vídeo procesado junto

```

for idx, (frame, usr_lms) in enumerate(frames):
    ref_lms = np.array(aligned_reference_coords[idx])
    aligned_ref_px = cv2.transform((ref_lms[:, :2]*[w,h])[None,...], A)[0]
    aligned_ref = aligned_ref_px / [w, h]

    diffs = np.linalg.norm(usr_lms[:, :2] - aligned_ref, axis=1)
    colors = [(0,255,0) if d<0.08 else (0,255,255) if d<0.13 else (0,0,255) for d in diffs]

    for parte, indices in BODY_PARTS.items():
        parte_diffs = [diffs[i] for i in indices]
        diffs_por_parte[parte].extend(parte_diffs)
        if any(d > 0.25 for d in parte_diffs):
            errores_por_parte[parte].append(round(idx/fps, 2))

        # Agregar similitud por frame para esa parte
        sim = penalized_similarity(parte_diffs)
        frame_similarities[parte].append(round(sim * 100, 2))

```

con sus landmarks del usuario. Para cada cuadro, primero se alinean las coordenadas detectadas con las del vídeo de referencia usando la transformación afín A, normalizando después por el tamaño de la imagen. Luego, se calculan las diferencias (diffs) entre los puntos clave del usuario y los de referencia mediante la norma euclidiana. Estas diferencias se convierten en colores (verde, blanco o rojo) según el nivel de precisión del movimiento. Después, para cada parte del cuerpo definida en BODY_PARTS (cabeza, torso, brazos, piernas), se extraen las diferencias correspondientes y se registran tanto en diffs_por_parte como en errores_por_parte si superan un umbral (0.25). Finalmente, se calcula una medida de similitud penalizada (penalized_similarity) para esa parte del cuerpo y se guarda en frame_similari-

ties, expresada en porcentaje por cuadro.

- Procesar y comparar

```

for i, lm in enumerate(usr_lms):
    # Color estándar basado en distancia
    color = (0, 255, 0) if diffs[i] < 0.08 else \
            (0, 255, 255) if diffs[i] < 0.13 else (0, 0, 255)

    # Si es la cabeza (landmark 0) y su similitud es mala (> 0.25), forzar color rojo
    if i == 0 and diffs[i] > 0.25:
        color = (0, 0, 255)

    # Dibuja el punto
    cv2.circle(frame, (int(lm[0]*w), int(lm[1]*h)), 6, color, -1)

for a,b in mp.solutions.pose.POSE_CONNECTIONS:
    col = (0,255,0) if colors[a]==colors[b]==(0,255,0) else \
          (0,255,255) if colors[a]==colors[b]==(0,255,255) else (0,0,255)
    cv2.line(frame,
             (int(usr_lms[a,0]*w), int(usr_lms[a,1]*h)),
             (int(usr_lms[b,0]*w), int(usr_lms[b,1]*h)),
             col, 2)
writer.write(frame)

writer.release()

```

Se recorren todos los landmarks (usr_lms) detectados en el usuario.

A cada punto se le asigna un color según la distancia (diffs[i]) entre el movimiento del usuario y la referencia:

Verde (precisión muy buena, <0.08).

Amarillo (aceptable, <0.13).

Rojo (error, ≥ 0.13).

Caso especial: si el punto corresponde a la cabeza (landmark 0) y la diferencia es mayor a 0.25, se fuerza a rojo.

Con cv2.circle se dibuja cada punto en el frame.

Luego, usando las conexiones de pose de Mediapipe (POSE_

CONNECTIONS), se dibujan las líneas entre los puntos del esqueleto. El color de la línea depende de los puntos que conecta (si alguno está en error, la línea también se pinta de rojo).

Finalmente, el frame con los dibujos se escribe en el archivo de salida con `writer.write(frame)`.

Al terminar, `writer.release()` guarda y cierra el vídeo.

- **Penalizador**

```
sims = [penalized_similarity(diffs_por_parte[p]) for p in BODY_PARTS]
similarity = np.clip(np.mean(sims), 0, 1)

analysis_path = output_path.replace('.mp4', '_analysis.txt')
with open(analysis_path, 'w') as f:
    for parte in BODY_PARTS:
        sim_pct = round(penalized_similarity(diffs_por_parte[parte]) * 100, 2)
        errores_crudos = errores_por_parte[parte]
        errores_filtrados = agrupar_errores(errores_crudos)
        if errores_filtrados:
            tiempos = ", ".join([segundos_a_timestamp(s) for s in errores_filtrados])
            f.write(f"{parte}: {sim_pct}% de similitud, errores en {tiempos}\n")
        else:
            f.write(f"{parte}: {sim_pct}% de similitud, sin errores destacados\n")
```

Calcula la similitud promedio por parte del cuerpo (cabeza, torso, brazos, piernas).

Saca la media y la limita entre 0 y 1 - esta será la similitud general del intento

Genera un archivo .txt paralelo al vídeo de salida, donde se guardará el análisis detallado de errores y porcentajes de similitud.

Para cada parte (cabeza, torso, brazos, piernas):

Calcula el % de similitud.

Busca los errores detectados (frames con diferencia alta).

Agrupar errores por tiempo (`segundos_a_timestamp`).

Si hubo errores - escribe:

cabeza: 85.4% de similitud, errores en 00:12, 00:25, 01:03

Si no hubo errores - escribe:

torso: 95.2% de similitud, sin errores destacados

- Guardar similitudes

```
# Guardar similitudes por parte y por frame en JSON
with open("static/frame_similarities.json", "w") as f:
    json.dump(frame_similarities, f)

fixed_output = output_path.replace('.mp4', '_fixed.mp4')
print("Iniciando ffmpeg...")
result = subprocess.run([
    'ffmpeg', '-y', '-i', output_path, '-i', input_path,
    '-c:v', 'libx264', '-crf', '23', '-preset', 'veryfast',
    '-c:a', 'aac', '-b:a', '192k',
    '-map', '0:v:0', '-map', '1:a:0?', '-pix_fmt', 'yuv420p', fixed_output
], stdout=subprocess.PIPE, stderr=subprocess.PIPE)
print("Finalizó ffmpeg.")

if result.returncode != 0:
    print("Error al ejecutar ffmpeg:")
    print(result.stderr.decode())
    raise RuntimeError("ffmpeg falló durante la generación del video final.")

return similarity, fixed_output, analysis_path, "static/frame_similarities.json"

if __name__ == '__main__':
    app.run(debug=True)
```

Este bloque cierra la función generando salidas y ensamblando el vídeo final. Primero serializa a disco las similitudes por parte y por fotograma en `static/frame_similarities.json` (`json.dump(frame_similarities, f)`), para que el frontend pueda graficarlas. Luego define `fixed_output` (misma ruta que `output_path` pero con sufijo `_fixed.mp4`) y ejecuta FFmpeg mediante `subprocess.run`: toma como entrada el vídeo procesado (primer `-i`, `output_path`) y el vídeo original (segundo `-i`, `input_path`), codifica el vídeo con H.264 (`-c:v libx264, -crf 23, -preset veryfast, -pix_fmt yuv420p`) y mapea el vídeo del procesado (`-map 0:v:0`) con el audio del original (`-map 1:a:0`) para producir `fixed_output` con audio conservado y sobreimpresiones dibujadas. Si FFmpeg devuelve un código distinto de cero, imprime el error estándar y lanza una `RuntimeError`. Finalmente, la función retorna: `similarity` (similitud global), la ruta del vídeo final `fixed_output`, la ruta del informe de texto `analysis_path` y la ruta del JSON de similitudes por frame. Fuera de la función, el bloque `if __name__ == '__main__': app.run(debug=True)` inicia el servidor Flask en modo debug para ejecutar la aplicación web localmente.

Conclusiones

Grado de eficiencia de la propuesta de solución interactiva

La propuesta de solución interactiva desarrollada en Dancistic ha demostrado un alto grado de eficiencia en la detección y análisis de similitudes entre coreografías. La implementación del modelo basado en MediaPipe y la sustitución de la visualización del esqueleto por un análisis narrativo detallado por

partes del cuerpo han permitido entregar resultados más claros y útiles para el usuario. El sistema funciona de forma fluida, con un procesamiento optimizado y un flujo de uso intuitivo, cumpliendo con las exigencias técnicas y de experiencia de usuario establecidas en la etapa de diseño. No obstante, se reconoce que el sistema aún presenta ocasionalmente fallas técnicas que requieren corrección y optimización, por lo que su desarrollo continuará para mejorar la estabilidad y confiabilidad en distintos escenarios de uso.

Nivel de logro de los objetivos

Se han cumplido los objetivos generales y específicos planteados al inicio del proyecto:

Objetivo general:

Se desarrolló un sistema interactivo capaz de analizar y entregar retroalimentación personalizada sobre la ejecución de coreografías, brindando al usuario información clara para mejorar su rendimiento.

Objetivos específicos:

Se implementó un mecanismo para la carga y comparación de vídeos en la plataforma.

Se optimizó el procesamiento mediante el cambio a MediaPipe, reduciendo los tiempos de espera y mejorando la accesibilidad.

Se diseñó un análisis por partes del cuerpo, con porcentajes de similitud y detección de errores por segundos específicos.

Se incorporaron mejoras sugeridas en pruebas de usuario, como la reproducción en bucle desde velocidad reducida hasta la normal y la visualización gráfica de los resultados.

Planificación y Proyección Nuevas Etapas de Desarrollo

- Para las siguientes etapas del proyecto, se contempla:
- Ampliar el catálogo de coreografías, incluyendo estilos más allá del K-pop.
- Implementar retroalimentación en tiempo real para el aprendizaje, similar a videojuegos como Dance Central.
- Generar alianzas con academias y profesores para integrar sus contenidos a la plataforma.
- Expandir el uso del modelo de análisis a otros rubros, como el deporte, la rehabilitación física y el entrenamiento personalizado.
- Incorporar métricas avanzadas y gamificación para incrementar la motivación de los usuarios y la retención en la plataforma.

En conjunto, estas proyecciones buscan consolidar a Dancistic como una herramienta versátil, adaptable y con potencial de crecimiento en múltiples áreas, manteniendo su enfoque en la interacción, la precisión y la experiencia del usuario, mientras se trabaja de forma continua en la solución de fallos técnicos y en la optimización de su rendimiento.

Bibliografía

Referentes académicos

- Aprendizaje motor y retroalimentación: El estudio de García y Sánchez (2021) analiza cómo la retroalimentación influye en la adquisición y perfeccionamiento de habilidades motoras, aportando fundamentos para el uso de reportes de rendimiento en tiempo real dentro de Dancistic.
Referencia completa:
García, J., & Sánchez, M. (2021). Aprendizaje motor y feedback. Dialnet. <https://dialnet.unirioja.es/servlet/articulo?codigo=7724427>
- Tecnología y educación en danza: Investigaciones como la de López y Martínez (2020) abordan la incorporación de herramientas tecnológicas en la enseñanza de la danza, evidenciando el potencial de entornos digitales para complementar el aprendizaje presencial.
Referencia completa:
López, A., & Martínez, P. (2020). La tecnología y la educación en danza. Dialnet. <https://dialnet.unirioja.es/servlet/articulo?codigo=8039069>
- Procesos formativos en K-pop dance cover: El análisis de Fernández (2022) sobre el aprendizaje y práctica del K-pop dance cover permite comprender el contexto cultural donde se inserta Dancistic, así como las dinámicas de comunidad y autoaprendizaje propias de este género.
Referencia completa:
Fernández, M. (2022). Análisis pedagógico de los procesos formativos del K-pop dance cover. Universidad de Chile.

- Entrenamiento y uso de tecnologías de análisis de movimiento: Estudios sobre aplicaciones de visión por computadora y análisis biomecánico, como el trabajo de Ramírez et al. (2018), sirven como referente técnico para la elección de librerías como MediaPipe y algoritmos de comparación de secuencias.
Referencia completa:
Ramírez, C., Soto, L., & Morales, F. (2018). Aplicaciones de análisis de movimiento en el entrenamiento deportivo. *Revista Chilena de Ciencias del Deporte*, 16(2), 45-58.
- Fuentes complementarias sobre tecnología y expresión corporal: La revisión de literatura en el documento de Fuentes (2017) sobre herramientas tecnológicas en la enseñanza de artes escénicas refuerza la idea de integración entre creatividad y tecnología.
Referencia completa:
Fuentes, E. (2017). Tecnología y educación artística: nuevas posibilidades para la expresión corporal. Universidad Nacional de Educación a Distancia.

Referentes primarios

Entrevistas a bailarines y coreógrafos de K-pop: Se realizaron 10 entrevistas a participantes de distintos niveles (básico, intermedio y avanzado) en Chile, entre abril y mayo de 2025. Estas entrevistas exploraron las principales necesidades, dificultades y expectativas de los usuarios potenciales, y sirvieron como base para el diseño de la interfaz y la definición de funcionalidades clave.

Observaciones de testeo del prototipo: Durante la fase de pruebas internas, se recopilaron datos de interacción con el sistema,

evaluando precisión del análisis, comprensión del reporte textual y aceptación de las funciones implementadas (reproducción en bucle a diferentes velocidades, gráficos comparativos, etc.). Estas observaciones guiaron las redefiniciones y ajustes técnicos de la propuesta.

Anexos

En esta sección se presentarán los videos y carpetas complementarias al desarrollo de este trabajo.

- Anexo 1: Video de presentación Dancistic
- Anexo 2: Carpeta del proyecto codificado.

Dancistic

Nicole Lorena Vargas Patarroyo



IV. ANEXOS

**ANEXO 1. Autorización única para todo Trabajo de Titulación presentada en
Universidad Gabriela Mistral**

I. Detalle de la memoria

Nombre de los integrantes:	Nicole Lorena Vargas Pataraco
Carrera / Magíster/Posgrado :	Diseño en Interacción Digital
Correo electrónico:	Nicole.Vargas@gmail.com
Título de la memoria:	Dancistic
Facultad:	Facultad de Arquitectura, Diseño y Artes Digitales
Mención a la que opta:	Mención en Interacción Digital
Profesor (es) guía(s):	Raimundo Vargas
Materias o descriptores: Asignar conceptos específicos al tema desarrollado en la memoria (máximo 4)	Danza, Tecnología, Innovación
Fecha de entrega a biblioteca (día, mes, año):	<u>05 / 12 / 2025</u>

pk

II. Autorización de publicación


Autorización para subir documentos digitalizados en el Repositorio Digital UGM

Autorizo a la Universidad Gabriela Mistral para incluir esta memoria en el repositorio digital de la universidad, con el fin de disseminarlo, almacenarlo y preservarlo, quedando disponible en texto completo y de libre acceso.

A considerar para la autorización de publicación de memorias en el repositorio digital:

- **Si usted como autor o autores deciden autorizar.** El documento estará disponible en texto completo.
- **Si usted como autor o autores deciden usar embargo.** Consiste en establecer una cantidad de tiempo en que la publicación no estará en libre acceso; es una alternativa a la publicación inmediata. El tiempo de embargo consiste en 6 meses, 1 año o 2 años.
- **Si usted como autor o autores deciden no autorizar.** Se creará un documento con datos básicos **portada, tabla de contenido y resumen** (metadatos) y colocará esa información en el sitio web, tal como si fuese una referencia bibliográfica. Se indicará en el registro que el autor decidió no autorizar el acceso al documento en texto completo.

Firma de autorización por cada uno de los autores de la memoria

		
Uso de embargo	Autorización total	No autorizo

Indique período de tiempo de embargo:

6 meses ____ 1 año ____ 2 años ____