



This is the **published version** of the bachelor thesis:

Valencia Pallo, Francisco Sebastián. *Herramienta de estimulación sensorial para niños y adultos con discapacidades múltiples..* Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/326539>

under the terms of the  license.

Herramienta para el estímulo sensorial de niños y adultos con pluridiscapacidad

Sebastian V.Pallo, *Estudiante de Ingeniería Informática e Ingeniería Electrónica de Telecomunicaciones*

Abstract— Las personas con pluridiscapacidad presentan graves limitaciones de comunicación e interacción con el entorno, acentuadas por modelos educativos poco flexibles y la escasez de herramientas tecnológicas adaptables y accesibles. Este trabajo propone el diseño y validación de un sistema interactivo basado en *pose estimation* e Inteligencia Artificial que permite transformar el movimiento corporal mínimo en una forma de interacción autónoma con el entorno. Empleando tecnologías *open source* y de bajo coste (Raspberry Pi, COCO, MediaPipe, redes neuronales convolucionales), se desarrolla un minijuego musical donde los *keypoints* corporales detectados por cámara se traducen en estímulos sonoros en tiempo real. El proyecto, validado con usuarios de la Fundación Nexa, demuestra la viabilidad de utilizar visión artificial como mecanismo no invasivo para fomentar la estimulación sensorial, el desarrollo motor y la autonomía en niños con pluridiscapacidad, alineado con los principios del Diseño Universal de Aprendizaje.

Index Terms— Pluridiscapacidad, Inteligencia Artificial, Pose Estimation, Visión Artificial, Tecnologías Asistivas, Interacción No Invasiva, Diseño Universal de Aprendizaje, Raspberry Pi, Keypoints, Heatmaps.



1 INTRODUCCIÓN

Las personas con pluridiscapacidad son aquellas que poseen de forma simultánea dos o más discapacidades a la vez. Las cuales, sumadas a un conjunto de patologías, tanto cognitivas como físicas, que afectan diariamente a su salud, originan un obstáculo considerable en la capacidad de percepción, comunicación y autonomía [1]. Aspecto que se intensifica con modelos educativos tradicionales, donde prevalecen metodologías sólidas, poco flexibles, que impiden el desarrollo académico y personal del niño [2]. Fomentando así, aún más, una dependencia de los familiares, amigos y personal cualificado.

Entorno que, generalmente, expresa una falta de sincronía entre las propuestas de los profesionales y las realidades del día a día en el ámbito cotidiano [3]. Principalmente, en la escasez de ayudas, así como de herramientas económicas y adaptables a las capacidades características del niño o adulto.

Motivo por el cual surge este trabajo en colaboración con la Fundación Nexa [4], entidad dedicada a mejorar la calidad de vida de los niños y niñas con pluridiscapacidad, permitiendo corroborar e identificar las necesidades más relevantes a cubrir, extraer los requisitos adecuados para desarrollar los sistemas pertinentes y validar los diseños a través de usuarios reales.

Dado el marco de cooperación, el proyecto en lugar de cubrir las necesidades más básicas, aseo, comida o movilización, tiene el objetivo de adaptar la vivienda a través del empleo de tecnologías. Alejando así los sentimientos de abandono al acogerlos en centros especializados, y contribuyendo al bienestar emocional de las familias [5].

En este caso, la interacción con el entorno propuesta se basa en la estimulación, mayormente sensorial, a través de

sistemas básicos de acción-reacción, empleando minijuegos que permiten alcanzar un grado de autonomía superior [3], desarrollando habilidades motrices, cognitivas y sociales. Siguiendo así los principios del Diseño Universal de Aprendizaje [6], proporcionando soluciones tecnológicas accesibles [2], inclusivas, económicas y de *open source* [7].

Para este caso de estudio, considerando la magnitud del trabajo y el tiempo establecido para su elaboración, se presenta el diseño, desarrollo y validación de una solución concreta, dando la posibilidad a los niños de tocar un instrumento sin ayuda externa, y con independencia de su capacidad motora. Se trata de un sistema cuya arquitectura es fácilmente escalable al propósito inicial, permitiendo incorporar más minijuegos y dispositivos, donde la interacción con el producto se traduce en el control de diferentes sistemas del domicilio personal.

2 OBJETIVOS DEL PROYECTO

El objetivo general del proyecto es reforzar la independencia de las personas con discapacidad a través del estímulo sensorial, especialmente visual y auditivo.

Asimismo, proporcionar soporte a organizaciones, personal cualificado y familias mediante la elaboración de una herramienta, accesible e intuitiva, enfocada en aspectos lúdicos y de rehabilitación.

A un nivel más técnico, se centra en el desarrollo de un sistema interactivo capaz de detectar y capturar el movimiento articular del niño, aplicando tecnologías avanzadas, como la inteligencia artificial (IA) [8], y reflejarlo en la pantalla del dispositivo en tiempo real según el minijuego seleccionado, en este caso: tocar un instrumento.

La idea se fundamenta en el principio de los sistemas SAAC [9], aumentativos y alternativos, los cuales emplean

diferentes elementos como: imágenes, dibujos, palabras o gestos, para incrementar y compensar las dificultades de comunicación.

2.1 Objetivos específicos

A raíz del propósito mencionado, se descomponen las siguientes metas a cumplir:

1. Profundizar el concepto de pluridiscapacidad y sus consecuencias.
2. Investigar el funcionamiento de los componentes, así como su instalación y configuración, *datasheet* [10].
3. Estudiar el campo de la IA [8], específicamente el funcionamiento de las redes neuronales [11].
4. Aprender técnicas de detección de poses [12], de reconocimiento facial y ocular [13].
5. Desarrollar o implementar un sistema *pose estimation* [12], y/o *eye tracking* [13].
6. Examinar lenguajes de programación adecuados para la creación de juegos 2D.
7. Crear un minijuego interactivo procesando las coordenadas capturadas por el sistema elaborado.
8. Implementar diferentes categorías dentro de cada modalidad con finalidad de adaptarlo a cada niño según su movilidad.
9. Implementar un menú principal capaz de gestionar la selección del minijuego.
10. Validar el producto con usuarios de la Fundación Nexé [4].

3 ESTADO DEL ARTE

Los modelos tradicionales de comunicación aplicados a niños con múltiples discapacidades generan cierto grado de incertidumbre, sentimientos de frustración y errores en la atención o enseñanza, al realizar interpretaciones subjetivas de sus emociones o acciones. Por tanto, refuerza la existencia de una dependencia a la ayuda externa en tareas básicas cotidianas.

El estado del arte en la aplicación de Inteligencia Artificial a niños con discapacidad severa o discapacidad múltiple refleja un cambio de paradigma profundo respecto a las tecnologías asistivas tradicionales. Mientras que históricamente los dispositivos se diseñaban bajo el supuesto de que el usuario debía interactuar activamente con la tecnología mediante habla, gestos, escritura o atención visual, las investigaciones más recientes plantean el enfoque inverso: la tecnología debe aprender a interpretar las señales mínimas, involuntarias o residuales del niño.

Este cambio conceptual se apoya en el desarrollo de la computación afectiva y el sentido pasivo, donde el sistema no requiere que el niño ejecute una acción, sino que detecta automáticamente estados emocionales, cognitivos o intencionales a partir de señales fisiológicas, micro-movimientos o patrones de comportamiento.

En el caso extremo de niños con discapacidades intelectuales profundas y múltiples (PIMD), donde no existen canales comunicativos observables, Tanabe et al. proponen un concepto radical: utilizar IA para reconocer emociones a partir de la variabilidad del ritmo cardíaco (RRI)

combinada con señales motoras extremadamente débiles como la mirada, la respiración, el movimiento de lengua y la actividad muscular registrada mediante EMG. Este trabajo demuestra que la emoción puede inferirse a partir de correlaciones fisiológicas complejas imposibles de detectar por observación humana directa, utilizando sensores de baja carga física y modelos explicables como Random Forest.

Este planteamiento conecta directamente con revisiones más amplias sobre herramientas digitales basadas en IA para discapacidad múltiple, donde se expone que las tecnologías más eficaces no son aquellas que exigen habilidades al usuario, sino aquellas que se adaptan a sus capacidades residuales, como *eye-tracking*, reconocimiento de micro-movimientos, sensores corporales e interfaces multimodales.

En paralelo, Zdravkova et al. muestran que este principio se extiende a un ecosistema amplio de tecnologías AAC (Augmentative and Alternative Communication), NLP, visión artificial y reconocimiento de voz/gestos, donde la IA convierte cualquier forma alternativa de expresión — pictogramas, gestos capturados por cámara, reconocimiento visual del habla, texto escrito o imágenes — en comunicación estructurada. Plataformas como OTTAA, entrenadas con millones de frases generadas por usuarios, permiten generar lenguaje funcional con interacciones mínimas y adaptarse progresivamente al usuario.

De esta manera, la IA no solo interpreta estados internos cuando no hay comunicación posible, sino que también traduce formas no convencionales de expresión en lenguaje comprensible cuando existe alguna capacidad residual.

Este mismo principio se traslada al ámbito educativo mediante aplicaciones móviles basadas en Intelligent Tutoring Systems (ITS), donde la IA adapta contenido, ritmo y retroalimentación al perfil cognitivo del alumno. El estudio de Faiz & Fazil muestra, desde la perspectiva docente, que apps como AutoSpark, Khan Academy Kids o Luca & Friends mejoran atención, motivación, autonomía y reducen el estrés en niños con discapacidad intelectual gracias a la personalización algorítmica del aprendizaje. Estas aplicaciones generan datos continuos sobre el rendimiento del niño, permitiendo ajustar dinámicamente la enseñanza sin aumentar la carga del docente.

Las revisiones sobre herramientas asistivas personalizadas refuerzan esta idea, destacando el uso de modelos de aprendizaje automático, árboles de decisión, LSTM, reconocimiento gestual mediante sEMG y técnicas de data mining para construir rutas de aprendizaje personalizadas y sistemas adaptativos para trastornos del neurodesarrollo.

4 METODOLOGÍA Y PLANIFICACIÓN

La metodología se implementa a través del *software* YouTrack [14], en el cual se elabora el cronograma del proyecto en forma de diagrama de Gantt [15], que refleja una división por fases:

- Inicio. Definición del proyecto, especificando objetivos y requisitos generales.
- Ejecución. Desarrollo del sistema interactivo,

diferenciando entre el despliegue HW y el desarrollo SW.

- Cierre. Proceso de integración final, con sus correspondientes pruebas, y la documentación o elaboración de informes como entregables, *milestones* [16].

Así mismo, su correspondiente EDT [17], con finalidad de cumplir los objetivos determinados en el anterior apartado (ver apéndice A).

El desarrollo de la red neuronal se realiza bajo el entorno online Google Colab, aprovechando su infraestructura de GPU gratuita, crucial para el entrenamiento de redes neuronales. Esta elección permite reducir tiempos de entrenamiento, manejar *datasets* extensos vía Google Drive, y evitar problemas de configuración de entorno local.

5 DESARROLLO DEL SISTEMA INTERACTIVO

5.1 Dataset COCO

Se selecciona el *dataset* COCO, de todo el repertorio posible como: AMASS, SURREAL Dataset o HumanEva, dado que es uno de los conjuntos de datos de estimación de poses humanas, de *open source*, más conocidos y populares. Además, se emplea dentro de uno de los ejemplos, **imx500_pose_estimation_higherhrnet_demo.py**, definidos en la librería Picamera2 de Python, la cual permite interactuar con la cámara de Raspberry Pi.

La solución propuesta tiene el foco situado en el diseño e implementación de una red neuronal capaz de aprender un patrón y generalizar el *dataset* mencionado con objetivo de efectuar predicciones precisas.

Sin embargo, examinando la estructura y el contenido, se observa que el conjunto seleccionados queda determinado por objetos de tipo `pyCOCO` que contienen metadatos, como anotaciones o categorías, en lugar de una estructura tipo *pipeline* que permite entregar los datos al modelo establecido.

```

Anotaciones
Datapoint 0:
Keys disponibles: ['segmentation', 'num_keypoints', 'area', 'iscrowd', 'keypoints',
Mask: [[214.65, 472.45, 229.75, 410.97, 212.49, 409.89, 197.39, 398.02, 201.71, 371.
Area: 42850.7618
Num Keypoints: 10
Keypoints: [289, 208, 2, 305, 192, 2, 286, 195, 2, 343, 187, 2, 0, 0, 0, 376, 267,
isfpowd: 0

```

Fig. 1. Anotaciones de una imagen del conjunto de entrenamiento COCO, en forma de diccionario, con las correspondientes llaves.

Dentro del primer bloque se observa una disposición en forma de diccionario cuyas llaves establecen aspectos relevantes para la aplicación, como: la máscara de segmentación, el array de *keypoints* y la cantidad detectada.

Ahora bien, la primera únicamente delimita la región de la categoría, persona en el caso de estudio, clasificando cada píxel de la imagen según si pertenece o no a dicha área. No refleja un mapa de calor (*heatmap*) entorno a la marca objetivo, por lo que se busca una alternativa empleando la distribución consecutiva.

Cada punto de interés contenido es definido por tres parámetros: coordenada en el eje de abscisas, coordenada en el eje de ordenadas y un *flag* de visibilidad. El último define el cero como punto no visible o detectado, el uno como

etiquetado y no visible, y el dos como etiquetado y visible.

```
Categorias
Datapoint 0:
Keys disponibles: ['supercategory', 'id', 'name', 'keypoints', 'skeleton']
Nombre super-categoría: person
Nombre categoría: person
Keypoints: ['nose', 'left_eye', 'right_eye', 'left_ear', 'right_ear', 'left_shoulder',
Esqueleto: [[16, 14], [14, 12], [17, 15], [15, 13], [12, 13], [6, 12], [7, 13], [6, 7],
```

Fig. 2. Categorías de una imagen del conjunto de entrenamiento COCO, en forma de diccionario, con las correspondientes llaves.

Las clases varían respecto a las definidas por una aplicación de tipo segmentación. En este caso, pasan a ser 17 las distinciones que puede tener un determinado píxel, una por cada punto de interés: nariz, ojos, orejas, hombros, codos, muñecas, caderas, rodillas y tobillos.



Fig. 3. Categorías y anotaciones de una imagen del conjunto de entrenamiento COCO. Se muestran los *keypoints* detectados, las conexiones en forma de esqueleto y el área de la persona.

5.2 Red neuronal convolucional (CNN)

5.2.1 Preprocesamiento

Concluida la selección del *dataset* y su análisis preliminar, se procede con la fase de preprocesamiento del modelo de red neuronal, destacada por las transformaciones realizadas a las imágenes de entrada para asegurar un aprendizaje efectivo y obtener así resultados precisos. Además, se incluye la creación de *heatmaps* como estructuras representativas de los puntos clave, empleando diferentes colores para reflejar la intensidad en el área de interés.

En primer lugar, se redimensiona y normaliza el *input* mediante la función *normalize()*. La **resolución** se reduce a **128x128** y el **rango de píxeles** pasa de uno entero [0,255] a uno de punto flotante [0.0, 1.0] para obtener entradas más manejables y así favorecer una convergencia rápida. El límite superior queda reservado para la celda que contiene el *keypoint*.

Después, los **mapas de calor** se fijan a una dimensión aún más inferior, de **64x64**, utilizando las coordenadas proporcionadas por el array de *keypoints*, ver Figura 1, para su correspondiente escalado, *keypoints2heatmaps()*. Ahora bien, con objeto de reducir cómputo innecesario, únicamente se consideran los puntos etiquetados y visibles indicados por el tercer parámetro, *flag*.

```

Sin normalizar: X: 192, Y: 103
Normalizado: X: 38.4, Y: 30.875878220140514
Escalado a dim del heatmap: X: 19, Y: 15
Keypoint en formato salida 64x64: (19, 15)

```

Fig. 4. Proceso de escalado de un *keypoint* de la imagen original, a la normalizada o redimensionada, y al formato del *heatmap*.

Interesa que el resto de los píxeles pierdan peso al distanciarse del pico en cuestión, motivo por el cual se emplea una **distribución Gaussiana** probando diferentes valores de desviación.

$$P[X = x] = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Fig. 5. Fórmula de la función de densidad de la distribución normal.

Dicho concepto estadístico describe de qué forma se distribuyen los datos alrededor de un valor concreto, en el caso de estudio, los *keypoints*.

```

Heatmap: [[0.8894073 0.8960133 0.90230745 ... 0.6717964 0.6604724 0.6490795 ]
[0.8945808 0.90122527 0.907556 ... 0.67570406 0.6643142 0.6528552 ]
[0.8994246 0.9061051 0.9124701 ... 0.67936283 0.66791123 0.6563901 ]
...
[0.6093271 0.61385286 0.6181649 ... 0.4602433 0.45248532 0.44468012 ]
[0.5980984 0.60254073 0.6067733 ... 0.45176193 0.44414684 0.4364856 ]
[0.5868418 0.59120053 0.5953535 ... 0.44325942 0.43578777 0.42827067]]

```

Fig. 6. *Heatmap* de una imagen de entrenamiento con desviación = 20.

En el caso de estudio, la variable corresponde a las diferentes coordenadas de cada píxel del *heatmap*, mientras que la media es la dedicada al conjunto del *keypoint*. Por tanto, la diferencia refleja el resultado de la distancia entre cada punto con el de interés, perdiendo peso con valores cada vez superiores. Por otro lado, la desviación define la dispersión o anchura de la curva clásica de campana. Aspecto relevante en términos de configuración dado que valores cercanos a cero enfocan o distinguen mejor el pico máximo, dando más probabilidad a los píxeles cercanos. Sin embargo, su implementación repercute en un incremento de la complejidad de la tarea a realizar al requerir una mayor precisión del modelo.

5.2.2 Procesamiento y arquitectura

Se define una arquitectura **U-Net**, propia de modelos de **red neuronal convolucional**, diseñadas para el procesamiento de datos en forma cuadrangular, como imágenes. La estructura está compuesta por un **codificador** de cinco niveles, *MobileNetv2*, y un **decodificador**, *pix2pix*, que permiten mantener el número de características reduciendo el tamaño del conjunto de entrada, es decir, la tasa de muestreo. Mientras que el primer bloque se encarga de reducir la resolución de la imagen de entrada, a la vez que el modelo identifica patrones más amplios, el segundo recupera el formato a través de convoluciones transpuestas y concatenaciones de características correspondientes del codificador.

Para dicho objetivo se emplea una técnica de *transfer learning* basada en la extracción de características. La estrategia se caracteriza por entrenar únicamente las capas introducidas en la parte concluyente, evitando así un procesamiento repetitivo y masivo de parámetros ya adquiridos, para la correspondiente adaptación a la tarea de *pose estimation*. Se mantiene el aprendizaje con un pequeño matiz al sustituir el hito terminal, generalmente caracterizado por capas *global pooling* y *dense*.



Fig. 7. Diagrama de una arquitectura U-Net con un input de 572x572 reducido a una dimensión de 28x28. Y posteriormente reconstruido a una resolución de 388x388.

En este caso, siguiendo el ejemplo centrado en la segmentación de imágenes, se congelan todas las capas del decodificador. El resto del modelo U-Net, el *decoder*, ejecuta la etapa de *train* de forma estándar, actualizando y corrigiendo los pesos en cada época, según la función de pérdida establecida.

Model: "U-Net"			
Layer (type)	Output Shape	Param #	Connected to
input_layer_7 (InputLayer)	(None, 128, 128, 3)	0	-
functional_5 (Functional)	[(None, 64, 64, 96), (None, 32, 32, 144), (None, 16, 16, 192), (None, 8, 8, 576), (None, 4, 4, 320)]	1,841,984	input_layer_7[0]...
sequential_4 (Sequential)	(None, 8, 8, 512)	1,476,608	functional_5[0]...
concatenate_4 (Concatenate)	(None, 8, 8, 1088)	0	sequential_4[0]... functional_5[0]...
sequential_5 (Sequential)	(None, 16, 16, 256)	2,507,776	concatenate_4[0]...
concatenate_5 (Concatenate)	(None, 16, 16, 448)	0	sequential_5[0]... functional_5[0]...
sequential_6 (Sequential)	(None, 32, 32, 128)	516,608	concatenate_5[0]...
concatenate_6 (Concatenate)	(None, 32, 32, 272)	0	sequential_6[0]... functional_5[0]...
sequential_7 (Sequential)	(None, 64, 64, 64)	156,928	concatenate_6[0]...
concatenate_7 (Concatenate)	(None, 64, 64, 160)	0	sequential_7[0]... functional_5[0]...
conv2d_transpose_9 (Conv2DTranspose)	(None, 64, 64, 17)	2,737	concatenate_7[0]...
Total params: 6,502,641 (24.81 MB)			
Trainable params: 4,658,737 (17.77 MB)			
Non-trainable params: 1,843,904 (7.03 MB)			
Heatmaps orig: (32, 64, 64, 17)			
Predicciones: (32, 64, 64, 17)			

Fig. 8. Estructura reducida del modelo de red neuronal, con arquitectura U-Net.

Se verifica el comportamiento mencionado al presentar parámetros no entrenables, que repercuten en una reducción de la carga computacional.

Situando el foco en la capa de salida, se especifica un *output* en forma de mapas de calor, de dimensión 64x64, y un número de canal equivalente a la cantidad de *keypoints* de interés, diecisiete. Por parte de la función de activación, se fija una de tipo **sigmoide**, permitiendo así forzar valores coherentes con la representación seleccionada, normalizados a un rango de cero y uno.

Previo al entrenamiento se realiza una etapa de compilación, en la cual se configura el proceso de aprendizaje determinando ciertos parámetros como: el optimizador, la función de pérdida y la métrica de evaluación.

El primero se fija a la **Estimación Adaptativa de Momentos (ADAM)** dado que ajusta el *learning rate* para cada parámetro, permitiendo un aprendizaje eficiente especialmente en redes neuronales complejas.

El segundo considera el **error cuadrático medio (MSE)** como medida del error entre las predicciones del modelo y los valores reales, caracterizando de nuevo una distancia.

Por último, la métrica se establece al **error medio absoluto (MAE)**, en lugar del típico *accuracy* empleado en tareas de clasificación. Un valor bajo indica predicciones próximas a los resultados esperados, mientras que uno elevado sugiere un ajuste deficiente.

5.2.3 Posprocesamiento

Posteriormente, y como última etapa del modelo, se elaboran dos funciones clave para la visualización y validación del conjunto implementado, *displayKeypoints()* y *displayHeatmaps()*.



Fig. 9. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*. Característica extraída una vez realizada la distribución Gaussiana.

Utilizando como prueba la misma imagen base de la Figura 3, se obtiene una caracterización de los picos exacta, al menos a simple vista.

Tomando un caso completamente distinto del conjunto de entrenamiento, sin detección de persona, se respalda el buen funcionamiento de los métodos aplicados.

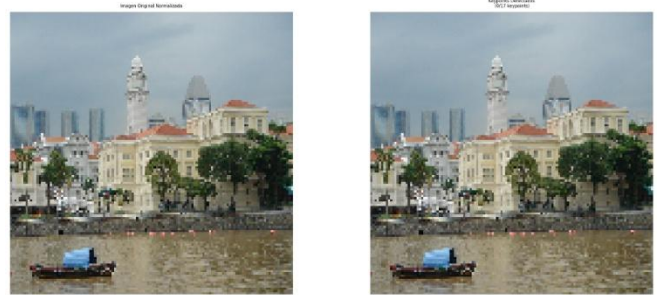


Fig. 10. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*, de una imagen sin persona detectada. Característica extraída una vez realizada la distribución Gaussiana.

Al no contener ningún individuo, el mapa de calor no refleja ningún pico, por defecto las coordenadas de puntos nulos quedan determinadas por el intervalo (0,0).

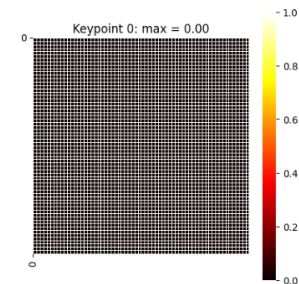


Fig. 11. *Heatmap* del *keypoint* 0, nariz, de la imagen sin persona detectada, *displayHeatmaps()*.

En cambio, al seleccionar una nueva muestra aleatoria del *dataset* en cuestión, se consiguen reflejar correctamente los picos, con el par de coordenadas correspondientes.



Fig. 12. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*, de una imagen aleatoria. Característica extraída una vez realizada la distribución Gaussiana.

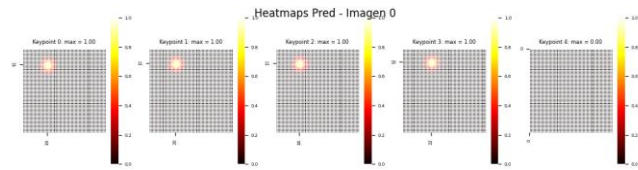


Fig. 13. *Heatmap* de los primeros cinco *keypoints*, de la imagen aleatoria, *displayHeatmaps()*.

5.3 Interfaz Gráfica

La etapa de diseño de la aplicación interactiva, con la interfaz gráfica propia de la cámara IA IMX500 de Sony, traslada el entorno de trabajo a la Raspberry pi 5 mediante una conexión remota empleando el programa RealVNC Viewer.

Esta fase se divide en tres bloques diferenciales: validación del modelo, diseño del menú de selección de *keypoints*, y la implementación de la mecánica musical.

Para elaborar una primera propuesta funcional, se emplea el modelo de red neuronal PoseNet dado el formato de uno de los **tensores de salida** (*keypoints*), tridimensional con distribución gaussiana, que posibilita la reutilización del código elaborado en la etapa anterior, de creación de una red neuronal propia. Las otras dos estructuras, *score* y *unknown*, varían en dimensión al contener más canales de los especificados por la tarea.

Así mismo, se utiliza como esquema base la *demo* predefinida en la librería **Picamera2**, nombrada *imx500_pose_estimation_higherhrnet_demo.py*.

Verificado el funcionamiento del modelo de referencia, se procede con el diseño del menú interactivo utilizando *pygame*, una librería ideal para el desarrollo de videojuegos, principalmente en 2D.

Como consecuencia, al integrar su propia ventana gráfica, entra en conflicto con la mostrada por Picamera2, evitando crear una dinámica de cambio de pantalla más visual y moderna. Motivo por el cual se realiza una integración en una única vista, distinguiendo el menú con un fondo transparente con objetivo de no perder visibilidad del comportamiento de la aplicación.

La selección se realiza con ayuda externa a través del ratón, pulsando el botón correspondiente al punto de interés.



Fig. 26. Ventana de visualización para una selección del *keypoint* 0, nariz.



Fig. 27. Ventana de visualización para una selección del *keypoint* 10, muñeca derecha.

Finalmente, se implementa la mecánica musical creando una estructura de directorios según los instrumentos escogidos: flauta, guitarra y piano, con opción a uno adicional. Cada una de las posibles melodías [] (do, re, mi, fa) es representada por una de las cuatro regiones verticales en que se divide la pantalla.

La dinámica consiste en la visualización de un menú complementario, destinado a los instrumentos, únicamente cuando el usuario de soporte elige la funcionalidad aislada de la estimación de poses. Para el caso generalizado no se otorga acceso al minjuego.

A partir de las coordenadas del tensor de salida de interés, correspondientemente preprocesadas, se identifica la zona que contiene el *keypoint* en cuestión, y, por tanto, el sonido a reproducirse.

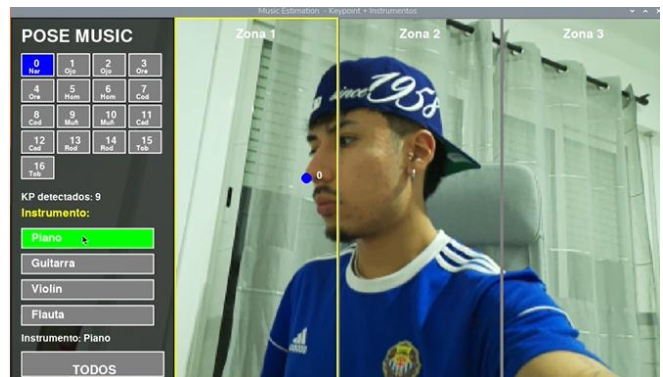


Fig. 26. Ventana de visualización para una selección del *keypoint* 0, nariz.

5.4 Icono de aplicación, ejecutable

Con finalidad de conseguir una operativa moderna, intuitiva y más visual, se crea un *script* a modo de sustitución de una ejecución *inline* por terminal.

```
[Desktop Entry]
Version=1.0
Type=Application
Name=Music Estimation
Comment=Instrumentos controlados por pose estimation
Exec=python3 /home/pi/Desktop/MusicEstimation/PoseNetMenuMusic.py
Icon=/home/pi/Desktop/MusicEstimation/Icon/IconMusicEstimation.png
Terminal=false
Categories=AudioVideo;Graphics;
StartupNotify=true
```

Fig. 26. *Script* elaborado para la creación de un ejecutable de tipo icono.



Fig. 26. Icono elaborado para la aplicación Music Estimation.

6 RESULTADOS

Las pruebas funcionales se consiguen a partir de la creación de un *pipeline*, con unos determinados parámetros de configuración, *batch_size* o *epochs*, para el proceso de las imágenes de forma eficiente.

En la primera fase de ejecución, caracterizada por funciones de tipo *print()* para la depuración del código, se observa la presencia de imágenes en escala gris, diferente al canal principal determinado por el patrón RGB, por lo que se requiere aumentar la dimensión de la imagen para el correcto funcionamiento del programa.

La visualización y comprobación del flujo especificado se consigue a partir de la extracción de la primera figura de dos *batches* consecutivos.



Fig. 12. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*, de las dos primeras imágenes de dos *batches* consecutivos. Característica extraída una vez realizada la distribución Gaussiana.

Al realizar las predicciones, se concluye que la función de pérdida no puede ser *binary_crossentropy* dado que acompañada de la métrica *accuracy* compara si cada píxel

del *heatmap* se encuentra por encima o por debajo de 0.5 respecto la etiqueta predicha. A raíz de la distribución gaussiana la mayoría de los píxeles se encuentran próximos a cero, únicamente el grupo minoritario cercano al *keypoint*, o pico, tendrá valores altos, cercano a uno.

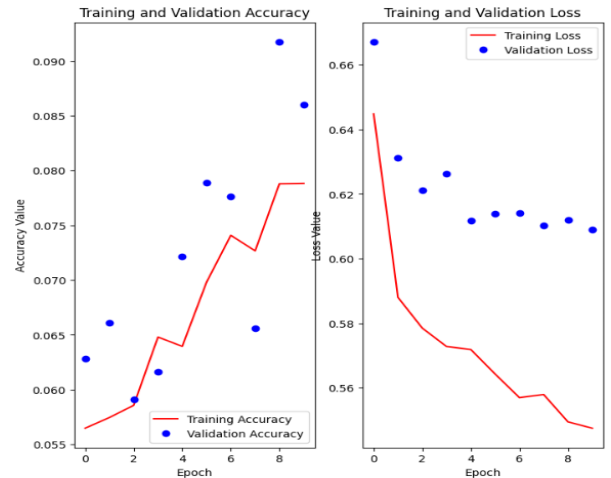


Fig. 13. Gráfica comparativa entre el acierto y la pérdida, tanto en la etapa de entrenamiento como en la de validación para la realización correcta de predicciones.

Motivo por el cual pasa a emplearse una pérdida más acorde a la aplicación de *pose estimation*, como el MSE.

Así mismo la validación pasa a reflejarse en el análisis de la curva de error y la inspección visual de los puntos del mapa de calor predicho por el modelo, en lugar del acierto u otra métrica poco comprensible.

Una posible métrica podría ser OKS/AP, distancia del *keypoint* original respecto al *keypoint* predicho (según las coordenadas).



Fig. 14. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*. Característica extraída una vez realizada la distribución Gaussiana.

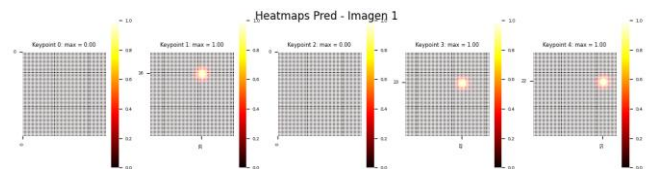


Fig. 15. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*. Característica extraída una vez realizada la distribución Gaussiana.

vez realizada la distribución Gaussiana.

Otra problemática recaer en la desviación de dicha distribución, los picos pueden estar relativamente dispersos originando un mapa muy difuso e impidiendo al modelo realizar un aprendizaje correcto. Cuanta más desviación, más aplanada resulta la representación gráfica.



Fig. 16. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*. Característica extraída una vez realizada la distribución Gaussiana.

Se puede observar en uno de los *heatmaps*, cuyo punto más distante aún tiene una probabilidad de ser pico del 42% aproximadamente. Interesa que se focalice el punto de interés dejando a los valores más alejados casi nulos.

Luego de modificar el parámetro mencionado a un valor de cinco, los mapas de calor pasan a diferenciarse drásticamente.

```
Sin normalizar: X: 389, Y: 381
Normalizado: X: 77.8, Y: 101.6
Escala a dim del heatmap: X: 38, Y: 50
Keypoint en formato salida 64x64: (38, 50)
```

Fig. 17. Proceso de escalado de un *keypoint* de la imagen original, a la normalizada o redimensionada, y al formato del *heatmap*, modificando la desviación de 20 a 5.

```
Heatmap: [[5.5316216e-35 2.4791008e-34 1.0674898e-33 ... 4.9027844e-27
1.9151596e-27 7.1877819e-28]
[4.0063942e-34 1.7955414e-33 7.7315797e-33 ... 3.5509455e-26
1.3871053e-26 5.2059231e-27]
[2.7879356e-33 1.2494660e-32 5.3801859e-32 ... 2.4710111e-25
9.6524706e-26 3.6226674e-26]
...
[2.5502519e-14 1.1429457e-13 4.9214734e-13 ... 2.2603294e-06
8.8294877e-07 3.3138033e-07]
[1.6099337e-14 7.2152089e-14 3.1068485e-13 ... 1.4269089e-06
5.5739059e-07 2.0919498e-07]
[9.7647415e-15 4.3762448e-14 1.8843953e-13 ... 8.6546402e-07
3.3807481e-07 1.2688305e-07]]
```

Fig. 18. *Heatmap* resultante de un *keypoint*, con una desviación de 5.

Pese a dicha alteración, los *keypoints* resultantes de la simulación, no concuerdan con la máscara creada y contenida en el conjunto de validación.

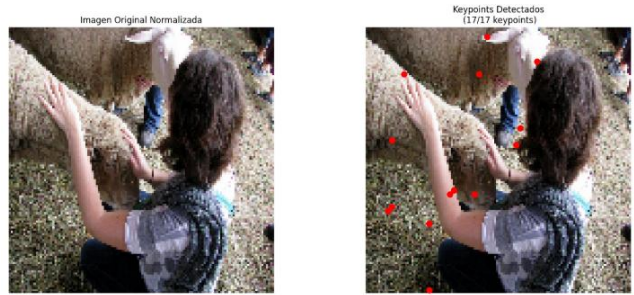


Fig. 19. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*. Característica extraída una vez realizada la distribución Gaussiana.

Resultado de la mala generalización, a raíz del no seguimiento de la tendencia de la métrica de prueba con la centrada en el entrenamiento.

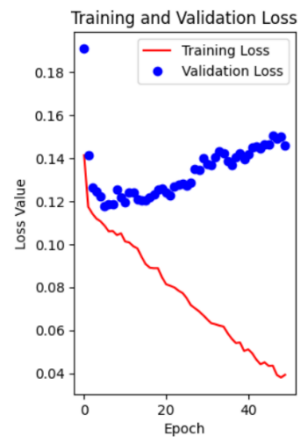


Fig. 20. Gráfica comparativa entre el acierto y la pérdida, tanto en la etapa de entrenamiento como en la de validación para la realización correcta de predicciones.

En lugar de decrecer con el número de épocas, a partir de la etapa 10, aproximadamente, el *val_loss* crece indefinidamente.

Examinando los mapas de calor, se observa que en el *dataset* el dedicado a la nariz no es visible, reflejando una estructura nula, mientras que el predicho identifica un pico en una determinada posición, dando lugar a un error en etapa de aprendizaje.

Cambiando la función de pérdida, a una focal caracterizada por ponderar en más cantidad los errores próximos al pico, se obtiene una mejora y aproximación notable, en términos de posición.

Así mismo, una función de activación de la última capa de *softmax*. Pero el problema está en que dicha función obliga a que las probabilidades dedicadas a cada píxel, de todos los mapas de calor, sumen un valor final de 1, por lo que no se obtiene independencia de canales para el correcto aprendizaje de picos.



Fig. 21. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*. Característica extraída una vez realizada la distribución Gaussiana.

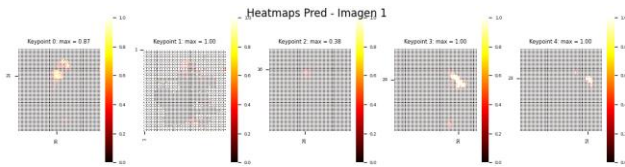


Fig. 22. Puntos de interés obtenidos a partir de los *heatmaps* individuales de cada clase, *displayKeypoints()*. Característica extraída una vez realizada la distribución Gaussiana.

Por parte de la pérdida, tampoco se obtuvo una mejoría.

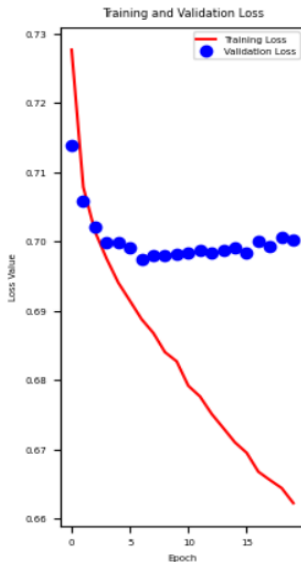


Fig. 23. Gráfica comparativa entre el acierto y la pérdida, tanto en la etapa de entrenamiento como en la de validación para la realización correcta de predicciones.

7 CONCLUSIÓN

El trabajo pone de manifiesto que, en personas con pluridiscapacidad, la principal barrera no es solo física o cognitiva, sino la dificultad de interacción y comunicación con el entorno, agravada por modelos educativos rígidos y poco adaptables. Frente a ello, el proyecto demuestra que es posible fomentar la autonomía no desde la asistencia directa, sino desde la adaptación tecnológica del entorno mediante estimulación sensorial e interacción lúdica, alineada

con el Diseño Universal de Aprendizaje y validada con usuarios reales de la Fundación Nexe.

A nivel técnico, se confirma la viabilidad de desarrollar soluciones avanzadas con tecnologías open source y de bajo coste (Raspberry Pi, COCO, Google Colab, MediaPipe), mostrando que la innovación no depende de hardware especializado sino de una correcta integración de herramientas existentes.

La implementación práctica de pose estimation revela aprendizajes clave: la importancia crítica del preprocesamiento del dataset COCO, la representación de keypoints mediante heatmaps gaussianos, la sensibilidad al parámetro de desviación, y la necesidad de utilizar funciones de pérdida adecuadas (MSE/focal) en lugar de métricas típicas como accuracy. Se evidencia que el reto principal no está tanto en la arquitectura (U-Net con transfer learning es válida), sino en cómo se formula el problema y se define el error.

Finalmente, el sistema demuestra que la pose estimation puede convertirse en un mecanismo de interacción no invasivo, donde el movimiento corporal mínimo detectado por cámara se traduce en control del entorno (en este caso, un minijuego musical), validando el principio central del estado del arte: usar la IA para interpretar las capacidades residuales del niño y convertirlas en autonomía funcional.

8 LA GERATIVA

Se ha utilizado *DeepSeek* para cuestiones relacionadas con el formato especificado por el estándar *IEEE Computer Society*.

También para aspectos de configuración del programa YouTrack [14], específicamente de calendario y dependencias, dado que en el repositorio de aplicaciones de la institución se ha sustituido el Microsoft Project [18] por el *software Planner* [19].

AGRADECIMIENTOS

REFERENCIAS

- [1] Nexe Fundació, “¿Qué es la Pluridiscapacidad?”, Nexe Fundació, disponible en <https://nexefundacio.org/es/que-es-la-pluridiscapacidad>. 1986.
- [2] Fundación ONCE, “Infancia con Discapacidad, Bienestar Emocional y Abandono Escolar Temprano,” pp. 7-9, disponible en <https://sid-inico.usal.es/documentacion/trazando-una-investigacion-inclusiva-propuestas-de-mejora-para-la-practica-educativa>. 2023.
- [3] M.Roca Roger, I.Úbeda Bonet, L.García Viñets and C.Caja López, “Parents Caregivers of Children with Multi Disabilities: A Qualitative Study about the Care and its Consequences,” Revista Española sobre Discapacidad Intelectual (Spanish Journal of Intellectual Disability), vol. 43 (4), no. 244, pp. 49-61, Dec 2012. (“Padres que Cuidan a sus Hijos con Pluridiscapacidad: Estudio Cualitativo sobre el Cuidado y sus Consecuencias”).
- [4] Nexe Fundació, “Nexe Fundació: Atención a la pluridiscapacidad”, Nexe Fundació, disponible en <https://nexefundacio.org/es>. 2025.
- [5] N.Morales Romo, A.Porras Montes y A.Huete García, “Estudio

sobre las familias con personas en situación de dependencia en España,” M.Hernando Ruiz, Fundación Caser, pp. 42-72, disponible en <https://sid-inico.usal.es/documentacion/estudio-sobre-las-familias-con-personas-en-situacion-de-dependencia-en-espana>. 2025.

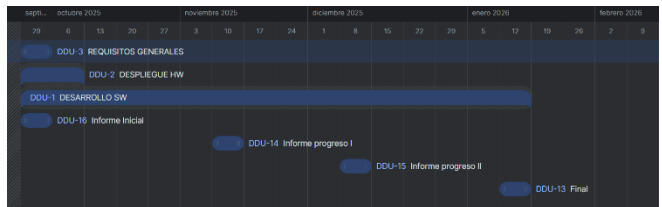
- [6] AFOE, “Los Principios DUA en Educación”, AFOE Formación, disponible en <https://www.afoe.org/dua-principios>. 2025.
- [7] Red Hat, “¿Qué es el Open Source”, Red Hat Official Website, disponible en <https://www.redhat.com/es/topics/open-source/what-is-open-source>. 2025.
- [8] IBM, “¿Qué es la Inteligencia Artificial o IA?”, IBM Think Topics, disponible en <https://www.ibm.com/mx-es/think/topics/artificial-intelligence>. 2025.
- [9] Plena Inclusión, “Sistemas Alternativos y Aumentativos de Comunicación. SAAC”, Plena Inclusión España, disponible en <https://www.plenainclusion.org/discapacidad-intelectual/recurso/sistemas-alternativos-y-aumentativos-de-comunicacion>. 2025.
- [10] Computer Hope, “Datasheet”, Computer Hope Jargon Dictionary, disponible en <https://www.computerhope.com/jargon/d/datasheet.htm>. 2025.
- [11] IBM, “¿Qué son las Redes Neuronales?”, IBM Think Topics, disponible en <https://www.ibm.com/es-es/think/topics/neural-networks>. 2025.
- [12] JRobotics, “¿Qué es la Pose Estimation y para qué sirve?”, JRobotics, disponible en <https://jrobotics.co/que-es-pose-estimation-y-para-que-sirve>. 2025.
- [13] Wikipedia, “Seguimiento de Ojos”, Wikipedia La enciclopedia libre, disponible en https://es.wikipedia.org/wiki/Seguimiento_de_ojos. 2025.
- [14] JetBrains, “YouTrack: Gestor de Proyectos y Seguimiento de issues”, JetBrains, disponible en <https://www.jetbrains.com/es-es/youtrack>. 2025.
- [15] OBS Business School, “¿Qué es un Diagrama de Gantt y para qué sirve?”, OBS Business School Blog, disponible en <https://www.obsbusiness.school/blog/que-es-un-diagrama-de-gantt-y-para-que-sirve>. 2025.
- [16] Asana, “Hitos de Proyecto: qué son y cómo identificarlos”, Asana Resources, disponible en <https://asana.com/es/resources/project-milestones>. 2025.n, *Linear Networks and Systems*. Belmont, Calif.: Wadsworth, pp. 123-135, 1993. (Book style)
- [17] Administrar Proyectos, “¿Qué es EDT y para qué sirve?”, Administrar Proyectos, disponible en <https://administrarproyectos.com/que-es-edt-y-para-que-sirve>. 2025.
- [18] https://github.com/raspberrypi/picamera2/blob/main/examples/imx500/imx500_pose_estimation_higherhr-net_demo.py#L55
- [19] <https://cocodataset.org/#format-data>
- [20] <https://www.tensorflow.org/tutorials/keras/classification?hl=es-419>
- [21] <https://www.tensorflow.org/tutorials/images/classification?hl=es-419>
- [22] <https://www.tensorflow.org/tutorials/images/segmentation?hl=es-419>
- [23] <https://www.geeksforgeeks.org/machine-learning/u-net-architecture-explained/>
- [24] Microsoft, “Instalar Project”, Microsoft Support, disponible en <https://support.microsoft.com/es-es/office/instalar-project-7059249b-d9fe-4d61-ab96-5c5bf435f281>. 2025.
- [25] Microsoft, “Microsoft Planner”, Microsoft 365, disponible en

<https://www.microsoft.com/es-es/microsoft-365/planner/microsoft-planner?msockid=3b91635c1ddb627c06f576a01c06638e>. 2025.

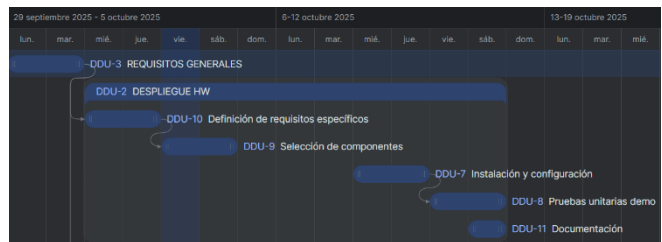
APÉNDICE

A ESTRUCTURA DE DESGLOSE DEL TRABAJO

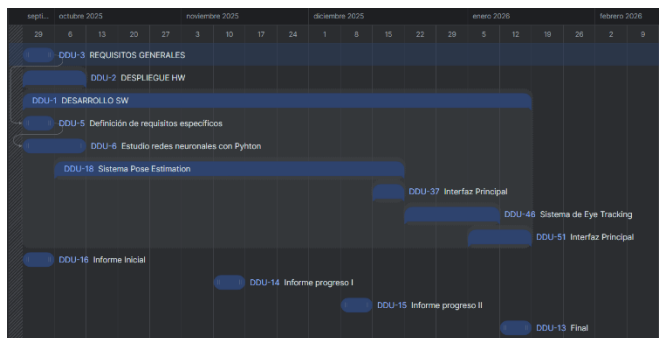
A.1 Planificación general



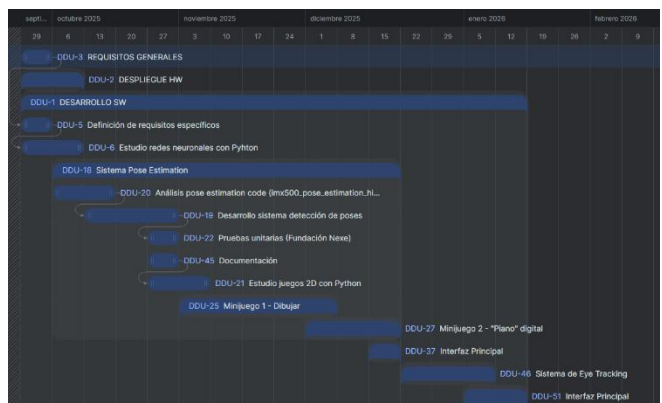
A.2 Despliegue del HW



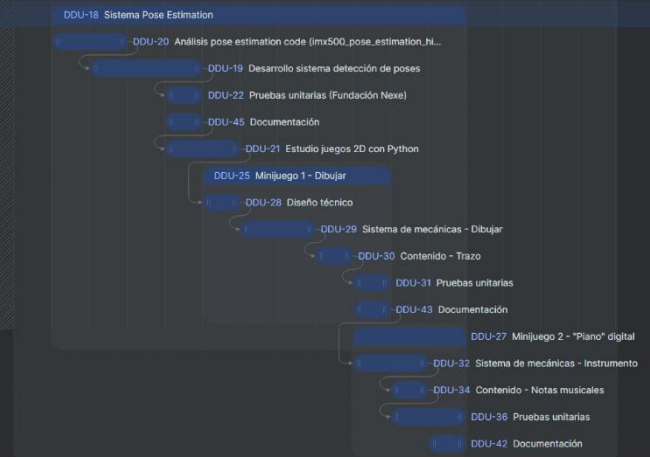
A.3 Desarrollo del SW



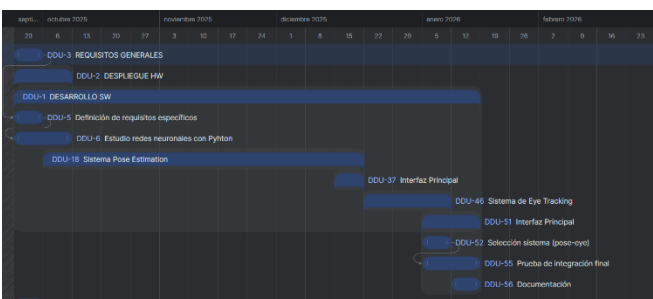
A.4 Desarrollo del SW: Sistema Pose Estimation



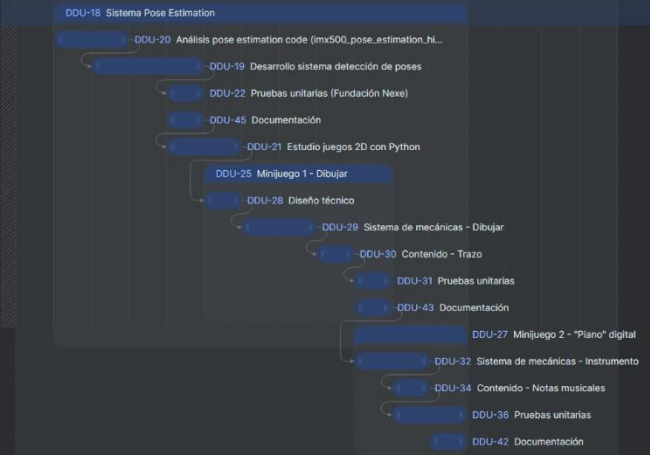
A.5 Desarrollo del SW: Minijuego 1 y Minijuego 2



A.8 Desarrollo del SW: Interfaz Principal v2



A.6 Desarrollo del SW: Interfaz Principal v1



A.7 Desarrollo del SW: Sistema Eye Tracking

