**Documento de Análisis y Diseño**

**de**

**MIRAGE**

Multiplatform Image Recognition and Gesture Experience.

**Fecha**  29/09/2025

**Versión** 1.0

**Integrantes** Norman Cortés

Vicente Flores

Gabriela González

Elías Martínez

# Introducción

## Diagnóstico e identificación del problema, desafío u oportunidad

El uso de interfaces tradicionales basadas en teclado, mouse o pantalla táctil limita la naturalidad y rapidez de interacción al manipular imágenes y/o videos en distintos dispositivos. Además, en contextos de presentación, trabajo colaborativo o entretenimiento, se requiere manipular imágenes sin necesidad de contacto físico con el dispositivo, ya sea por conveniencia o por razones de higiene.

El desafío consiste en crear una interfaz natural que permita manipular video mediante gestos de la mano frente a la cámara, reduciendo la fricción en el proceso y permitiendo la interacción intuitiva entre dispositivos.

## Propuesta de solución

La solución consiste en un **sistema de reconocimiento de gestos en tiempo real** que permite manipular la reproducción de videos dentro de un reproductor multimedia (ej. VLC, MPV, GStreamer) sin necesidad de interfaces gráficas adicionales.

El usuario podrá:

* **Seleccionar un video** mediante gesto definido.
* **Reproducir o pausar** con gesto de play/pause.
* **Navegar entre videos** mediante barridos laterales de la mano (izquierda/derecha).
* **Adelantar o retroceder** segmentos del video con movimientos rápidos.
* **Activar o desactivar sonido (mute/unmute)** con gesto dedicado.
* **Transferir videos entre dispositivos** conectados en red local mediante gestos de arrastre hacia un dispositivo remoto.

El desarrollo se apoyará en **bibliotecas de visión por computador** (OpenCV, MediaPipe) para el preprocesamiento, y en **modelos de aprendizaje profundo** (CNN + LSTM implementados en PyTorch/ONNX) para capturar tanto características espaciales como temporales de los gestos.  
La comunicación entre dispositivos se resolverá a través de **red local**, empleando protocolos como WebSocket o WebRTC para asegurar baja latencia y compatibilidad multiplataforma.

## Estado del Arte y de la Técnica

El reconocimiento de gestos ha sido explorado ampliamente en la literatura y en la industria, con enfoques que van desde **librerías open-source** como MediaPipe Hands hasta **soluciones comerciales** como Leap Motion. Sin embargo, la mayoría de estos desarrollos se orientan al control de interfaces en realidad virtual, videojuegos o entornos de realidad aumentada, y no específicamente al **control directo de reproductores multimedia ligeros y multiplataforma**.

A nivel académico, se han propuesto modelos híbridos que combinan **redes convolucionales (CNN)** para extracción de características espaciales con **redes recurrentes LSTM** para el análisis temporal de secuencias de gestos [Zhang et al., 2020], logrando alta precisión en tiempo real.

Por otro lado, soluciones industriales como **Air Gesture de Samsung** o **Microsoft Kinect** (ya descontinuado) demuestran la factibilidad del control gestual para interacción multimedia, pero presentan limitaciones de escalabilidad y dependencia de hardware específico.

## Referencias

 Zhang, F., et al. (2020). Hand Gesture Recognition Using Deep Learning and Computer Vision. IEEE Transactions on Multimedia.

 MediaPipe Documentation: https://developers.google.com/mediapipe

 OpenCV Documentation: https://docs.opencv.org/

# Requisitos del Sistema

## Requisitos Funcionales

1. Detectar manos en tiempo real mediante cámara integrada.
2. Interpretar gestos predefinidos (play/pause, siguiente, anterior, adelantar, retroceder, mute/unmute) y mapearlos a acciones específicas en el reproductor multimedia.
3. Permitir navegar entre videos mediante gestos de barrido lateral (izquierda/derecha), reproduciendo el video correspondiente de la lista.
4. Implementar la función de adelantar y retroceder dentro de un video en curso mediante gestos definidos de desplazamiento rápido.
5. Activar y desactivar el sonido (mute/unmute) con un único gesto dedicado que alterna entre ambos estados.
6. Mostrar feedback visual superpuesto en la pantalla del reproductor (íconos o indicadores) para confirmar la ejecución de gestos y estados del sistema (ej. video pausado, mute activado).
7. Transferir videos a otro dispositivo conectado en la red local mediante un gesto de arrastre hacia el borde de la pantalla en dirección al dispositivo destino.
8. Gestionar la comunicación entre dispositivos en red local para asegurar la transmisión de comandos y la transferencia de archivos con baja latencia y confiabilidad.

## Requisitos no funcionales

RNF1 Latencia de detección < 150 ms.

RNF2 Precisión de reconocimiento de gestos ≥ 95% en condiciones normales de iluminación.

RNF3 Procesamiento local de video (privacidad por defecto).

RNF4 Soporte multiplataforma (Windows/macOS/Linux en PC, Android/iOS en tablet/smartphone).

RNF5 Escalabilidad para manejar 1–3 dispositivos conectados.

## Requisitos de Interfaces

El sistema debe responder a una serie de **eventos externos generados por el usuario** a través de gestos manuales detectados por la cámara. Estos eventos se consideran como la interfaz de interacción entre el usuario y el sistema, y cada uno desencadena una respuesta específica del software.

Los gestos implementados corresponden a **comandos multimedia estándar** (reproducir, pausar, navegar, adelantar, retroceder, silenciar/activar sonido) y a la **transferencia de archivos en red local (LAN)**.

Las tablas siguientes enumera los eventos externos reconocidos y las respuestas esperadas del sistema.

Tabla 1: Eventos externos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evento** | **Descripción** | **Iniciador** | **Parámetros** | **Respuesta** |
| Selección | Gesto de seleccionar imagen | Usuario | Coordenadas de video | Video marcado |
| Reproducir/  Pausar | Gesto de play/pause | Usuario | idVideo | Estado de reproducción |
| Navegar | Barrido lateral (izq/der) para cambiar de video | Usuario | Dirección (izq/der) | Cambio de video |
| Adelantar/  Retroceder | Gesto de girar mano abierta | Usuario | |  | | --- | |  |  |  | | --- | |  |   idVideo, dirección (adelante/atrás) | Video adelantado/retrocedido |
| Mute/  Unmute | Gesto para silenciar o activar sonido (toggle) | Usuario | idVideo | Estado de sonido |
| Transferencia LAN | |  | | --- | |  |  |  | | --- | | Gesto de mover hacia dispositivo remoto | | Usuario | idVideo, dispositivoDestino | Imagen transferida |

Tabla 2: Respuestas del sistema

|  |  |  |
| --- | --- | --- |
| **Respuesta** | **Descripción** | **Parámetros** |
| Video marcado | El sistema resalta el video seleccionado | idVideo |
| Estado de reproducción | El video cambia entre reproducir/pausar | idVideo, estado |
| Cambio de video | Se muestra el video anterior o siguiente | dirección (izq/der) |
| Video adelantado | El video avanza unos segundos | idVideo, segundosAvance |
| Video retrocedido | El video retrocede unos segundos | idVideo, segundosRetroceso |
| Estado de sonido | El sistema alterna entre silencio y sonido activo | idVideo, estadoSonido (mute/unmute) |
| Video transferido | Video enviado a dispositivo remoto | idVideo, dispositivoDestino |

## Requisitos de Ambiente

### Hardware de Desarrollo

* **Procesador:** CPU multinúcleo (mínimo 4 núcleos, recomendado Intel i5 / AMD Ryzen 5 o superior).
* **Memoria RAM:** 8 GB (mínimo) / 16 GB (recomendado).
* **Cámara:** Cámara web HD (720p mínimo, 30 fps) o equivalente.
* **Almacenamiento:** 10 GB libres para instalación de contenedores y dataset de prueba.
* **Conectividad:** Red LAN o Wi-Fi para transferencia de archivos entre dispositivos.

### Software de Desarrollo

Python 3.10, OpenCV, MediaPipe, PyTorch/ONNX, WebSockets/WebRTC.

* **Sistema Operativo:** Windows 10/11, macOS 12+ o distribución Linux compatible con Docker.
* **Plataforma de contenedores:** Docker Engine + Docker Compose.
* **Lenguaje de programación:** Python 3.10 o superior.
* **Dependencias de visión por computador:** OpenCV, MediaPipe.
* **Framework de comunicación:** FastAPI (para APIs REST) + WebSocket.
* **Procesamiento multimedia:** VLC, MPV, GStreamer.

### Datasets e Imágenes/Videos de Prueba

* Datasets externos:
  + **EgoHands Dataset**: colección de imágenes en primera persona con manos etiquetadas, útil para detección y segmentación de gestos.
  + **Hand Gesture Recognition Dataset (Kaggle, HaGRID, Hand-gesture dataset, etc.)**: bases de datos que incluyen gestos comunes (play, stop, swipe, etc.) bajo diversas condiciones de luz y fondo.
  + **SHREC’17 Track – Hand Gesture Dataset**: gestos dinámicos capturados en secuencia, adecuados para modelos con LSTM.
* Material de prueba generado localmente:
  + **Videos cortos grabados con cámara web** en distintos entornos (luz natural, luz artificial, distintos usuarios) para validar robustez del modelo.
  + **Secuencias de gestos específicos del proyecto** (reproducir, pausar, siguiente, anterior, adelantar, retroceder, mute/unmute, transferir).
  + **Videos multimedia estándar** (mp4, avi) utilizados en pruebas de control sobre reproductores (ej. VLC, MPV).

## Perfiles de Usuario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Perfil** | **Socioeconómico y cultural** | **Ocupacional** | **Etario** | **Características físicas, fisiológicas, psicológicas.** | **Otros** |
| **Usuario doméstico** | Personas con acceso a dispositivos multimedia en el hogar. | Estudiantes, trabajadores remotos, usuarios de entretenimiento digital. | 18–50 años | Condiciones físicas generales, habilidades motoras básicas de la mano. |  |
| **Usuario académico** | Estudiantes universitarios y docentes con interés en nuevas interfaces. | Uso en clases o presentaciones interactivas. | 18–45 años | Buena familiaridad con herramientas digitales. |  |

# Planificación del Proyecto

## Objetivo General (Goal)

Desarrollar un sistema de manipulación de imágenes y video entre dispositivos mediante gestos naturales de la mano, sin necesidad de teclado, mouse, ni barra de acciones.

## Objetivos Específicos

O1. Diseñar e implementar un módulo de detección y reconocimiento de gestos en tiempo real.

O2. Implementar la manipulación de videos dentro de un dispositivo mediante gestos.

O3. Desarrollar la transferencia de videos entre dispositivos conectados en red local

## Actividades y Milestones

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Responsable** | **Participación**  **equipo** | **Fecha**  **comienzo** | **Fecha**  **término** | **% de**  **avance** |
| **[Nombre de elemento (objetivo, actividad de objetivo, actividad de gestión (no resuelve un objetivo en específico), tarea, milestone)]** | [Responsable de avance del elemento.] | [% de participación de los miembros del equipo en la actividad] | [Fecha de comienzo del elemento. Si aún no comenzado, poner fecha tentativa de comienzo ]. | [Fecha de término del elemento. Si no terminado, poner fecha tentativa de término] | [% de avance del elemento.] |
| **A1.1 Investigación y definición de gestos.** |  |  | 22/09/2025 |  |  |
| **A1.2 Investigación y definición de tecnologías** |  |  |  |  |  |
| **A1.3 Adquisición y preprocesamiento de datos** |  |  |  |  |  |
| **A1.4 Diseño y arquitectura del modelo** |  |  |  |  |  |
| **A1.5 Entrenamiento y optimización del modelo** |  |  |  |  |  |
| **A1.6 Implementación y encapsulamiento del módulo**  **A1.7 Pruebas y validación del módulo en tiempo real** |  |  |  |  |  |
| **H1.1 Prototipo de detección básica** |  |  |  |  |  |
| **H1.2 Conjunto de gestos implementado** |  |  |  |  |  |
| **A2.1 Investigación de librerías** |  |  |  |  |  |
| **A2.2 Definir acciones y diseñar mapeo** |  |  |  |  |  |
| **A2.3 Implementación del control** |  |  |  |  |  |
| **A2.4 Desarrollo de feedback visual** |  |  |  |  |  |
| **A2.5 Pruebas funcionales** |  |  |  |  |  |
| **H2.1 Primer control de reproducción** |  |  |  |  |  |
| **H2.2 Implementación de gestos adicionales** |  |  |  |  |  |
| **H2.3 Retroalimentación inmediata de gestos** |  |  |  |  |  |
| **A3.1 Diseño del protocolo de red local para comandos** |  |  |  |  |  |
| **A3.2 Empaquetado y configuración de instalación** |  |  |  |  |  |
| **A3.3 Prueba del sistema completo y reporte final** |  |  |  |  |  |
| **H3.1 Orquestación inicial con Docker Compose** |  |  |  |  |  |
| **H3.2 Configuración portable y parametrizable** |  |  |  |  |  |
| **H4.1 Envío básico de archivos** |  |  |  |  |  |
| **H4.2** |  |  |  |  |  |
|  |  |  |  |  |  |

## Carta Gantt

A chart with multiple colored squares

AI-generated content may be incorrect.

# Diseño de Arquitectura del Sistema

## Diagrama de Contexto

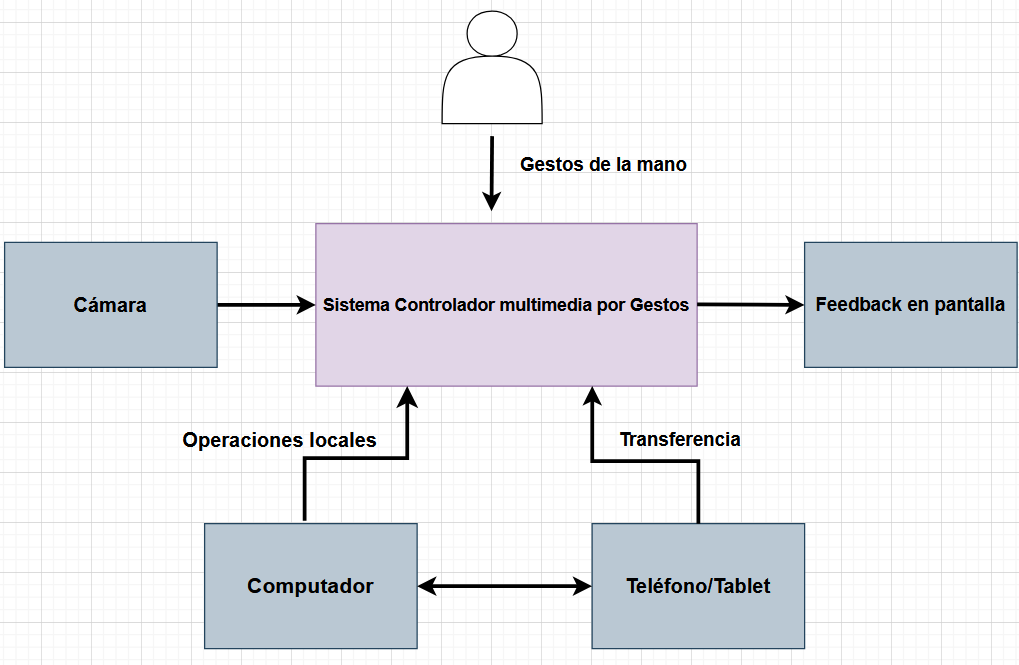


Figura 1: Diagrama de contexto del sistema

## Diagrama de Arquitectura

El diagrama debería mostrar **los módulos principales** y **los flujos de datos**.

**Módulos:**

* **M1 – Servicio de Gestos (SG):** Procesa la señal de la cámara, detecta y reconoce gestos usando CNN+LSTM.
* **M2 – Controlador Multimedia (CM):** Traduce los gestos reconocidos en comandos al reproductor (ej. VLC/MPV).
* **M3 – Servicio de Comunicación LAN (LAN):** Permite la transferencia de videos y sincronización de acciones entre dispositivos.
* **M4 – Base de Datos / Gestión de Videos (BD):** Mantiene registro de videos disponibles.

**Flujos:**

* Cámara → M1 → M2 → Reproductor multimedia.
* M1 ↔ M3→ Control remoto (envío de comandos o transferencia de videos).
* M4 ↔ M2 (para identificar videos y administrar la lista de reproducción).



Figura 2: Diagrama de arquitectura del sistema

## Enumeración de Módulos

La Tabla 3 muestra los módulos de la Figura 2. Por cada módulo se entrega un breve párrafo descriptivo de su propósito, además de la sección en donde se especifica el módulo en detalle.

Tabla 3: Módulos de la arquitectura del sistema

| **Módulo** | **Propósito** | **Sección** |
| --- | --- | --- |
| Servicio de Gestos (SG) | Captura las imágenes desde la cámara y aplica modelos de visión por computador (CNN+LSTM) para reconocer gestos en tiempo real. Sus resultados son enviados al Controlador Multimedia o al Servicio de Comunicación LAN. | 5 |
| Controlador Multimedia (CM) | Traduce los gestos reconocidos en comandos concretos para el reproductor de video (play, pause, siguiente, anterior, adelantar, retroceder, mute/unmute). Se conecta directamente con VLC/MPV u otro motor multimedia. | 5 |
| Servicio de Comunicación LAN (LAN) | Maneja el envío y recepción de comandos o archivos de video entre dispositivos en red local, permitiendo extender la manipulación más allá de un solo equipo. | 5 |
| Gestión de Videos (BD) | Administra los videos locales disponibles, guardando metadatos como nombre, ubicación y estado. Provee esta información al Controlador Multimedia. | 5 |

## Matriz de Requisitos Funcionales y Módulos

Reescriba los requisitos funcionales, con su abreviatura. La siguiente matriz muestra qué módulos del sistema implementan qué requisitos funcionales.

[*NOTA*: Una línea vacía en la matriz indica un requisito funcional no implementado; en ese caso es probable que falte un módulo. Una columna vacía indica un módulo no es útil para estos requisitos funcionales; en ese caso es posible que falte un requisito funcional. Una columna demasiado llena indica un módulo que participa en demasiados requisitos; esto podría ser síntoma de una mala modularización, porque un cambio en el módulo puede afectar a todos los requisitos. Algo similar ocurre con una fila demasiado llena, que corresponde a un requisito que se implementa en todas partes; un cambio en el requisito podría implicar cambios en todas los módulos.]

[*NOTA*: La matriz de abajo debe re-formatearse de modo que tenga columnas para todas los módulos y filas para todos los requisitos funcionales. Una vez re-formateada la matriz, ponga una X en una celda para indicar que el módulo de la columna respectiva ayuda a implementar el requisito funcional de la fila respectiva.]

Tabla 4: Matriz de requisitos funcionales y módulos

|  | M1 | M2 | M3 | M4 |
| --- | --- | --- | --- | --- |
| 1. Detectar manos en tiempo real mediante cámara integrada. | X |  |  |  |
| 1. Interpretar gestos predefinidos (play/pause, siguiente, anterior, adelantar, retroceder, mute/unmute). | X | X |  |  |
| 1. Navegar entre videos mediante gestos de barrido lateral. | X | X |  | X |
| 1. Adelantar y retroceder dentro de un video en curso mediante gestos definidos. | X | X |  |  |
| 1. Activar y desactivar el Sonido (mute/unmute) con un gesto dedicado. | X | X |  |  |
| 1. Mostrar feedback visual superpuesto en la pantalla del reproductor. | X | X |  |  |
| 1. Transferir videos a otro dispositivo en red local mediante gesto de arrastre. | X |  | X | X |
| 1. Gestionar la comunicación entre dispositivos en red local. |  |  | X |  |

# Módulo [XX]

[*NOTA*: Copie esta sección como plantilla para cada módulo del sistema.]

## Módulo Servicio de Gestos (SG)

|  |  |
| --- | --- |
| **Propósito** | Detectar y reconocer gestos de la mano en tiempo real utilizando la cámara del dispositivo, transformando las señales visuales en eventos de control que serán enviados al Controlador Multimedia o al Servicio LAN. |
| **General** | Este módulo se ubica al inicio de la cadena de procesamiento. Su entrada principal es el flujo de video de la cámara integrada, y su salida son etiquetas de gestos reconocidos (ej. play, pause, siguiente, anterior, adelantar, retroceder, mute/unmute, transferir). Se comunica directamente con el **Controlador Multimedia** (para control local del reproductor) y con el **Servicio LAN** (para transferencia o sincronización remota). |
| **Marco Técnico del Módulo** | * Visión por computador: OpenCV y MediaPipe para detección de manos y landmarks. * Modelos de IA: CNN para extracción de características espaciales y LSTM para secuencias temporales de gestos dinámicos. * Frameworks: PyTorch / ONNX para entrenamiento e inferencia. * Optimización: técnicas de normalización y data augmentation para mejorar robustez ante variaciones de iluminación, fondo y usuario. |

**5.2. Módulo Controlador Multimedia (CM)**

|  |  |
| --- | --- |
| **Propósito** | Traducir los gestos reconocidos en comandos concretos para el reproductor multimedia local (ej. VLC, MPV, GStreamer). |
| **General** | Se ubica en el centro de la arquitectura. Su entrada son los eventos de gestos provenientes del Servicio de Gestos, y su salida son acciones ejecutadas sobre el reproductor (play, pause, siguiente, retroceder, mute/unmute). Provee además feedback visual superpuesto en la pantalla para confirmar la acción al usuario. |
| **Marco Técnico del Módulo** | * **Reproductor multimedia:** VLC/MPV con APIs o bindings de Python. * **Gestión de comandos:** Traducción de gestos a atajos de teclado o llamadas a API. * **Feedback visual:** overlays generados con OpenCV o GStreamer. |
| **Estructura General** | 1. Recepción del gesto reconocido (ej. "swipe derecha"). 2. Traducción a comando multimedia ("video siguiente"). 3. Ejecución en el reproductor (llamada API). 4. Generación de feedback visual (icono o mensaje en pantalla). |

**5.3 Módulo Servicio de Comunicación LAN (LAN)**

|  |  |
| --- | --- |
| **Propósito** | Permitir la transferencia de videos y la sincronización de acciones entre distintos dispositivos conectados a la misma red local. |
| **General** | Este módulo se comunica con el Controlador Multimedia y con otros nodos en la red. Recibe como entrada eventos de transferencia (ej. arrastrar hacia el borde) o comandos de control remoto, y su salida es la ejecución de la acción en el dispositivo destino. |
| **Marco Técnico del Módulo** | * **Protocolos de red:** WebSocket para baja latencia, WebRTC para streaming y sincronización. * **Descubrimiento de dispositivos:** Service Discovery (ej. mDNS). * **Serialización de datos:** JSON para comandos, chunks binarios para archivos de video. |
| **Estructura General** | 1. Escucha de eventos de transferencia desde el Servicio de Gestos 2. Serialización y envío de comando/video al dispositivo remoto. 3. Recepción y confirmación de entrega. 4. Ejecución del comando en el reproductor del dispositivo destino. |

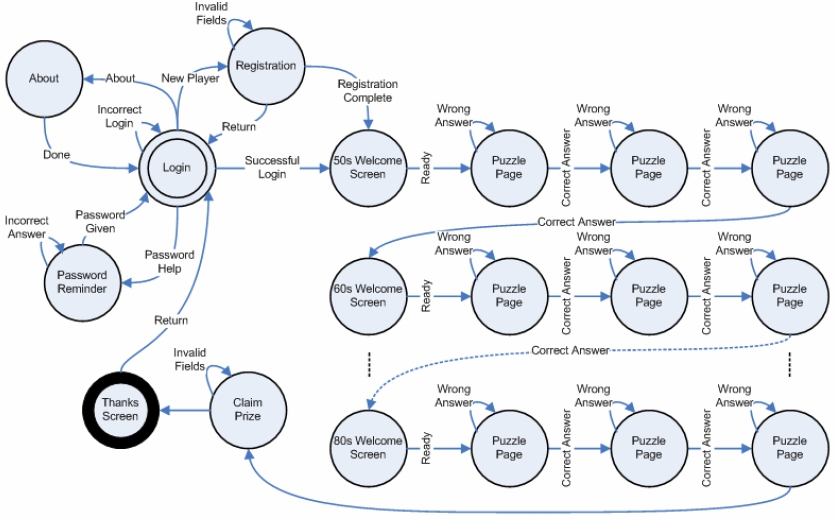
**5.4 Módulo Gestión de Videos (BD)**

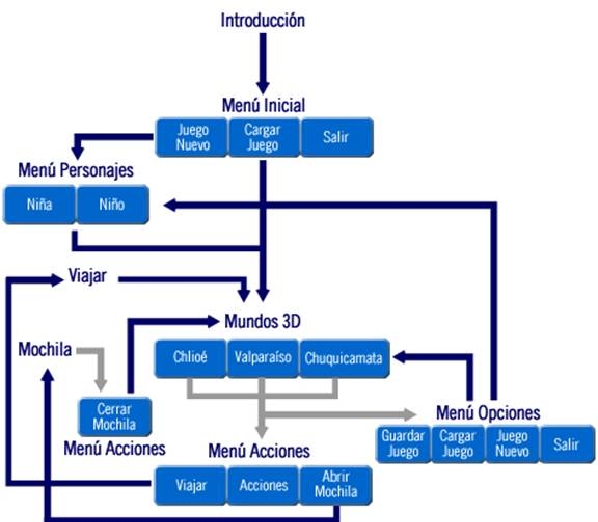
|  |  |
| --- | --- |
| **Propósito** | Administrar los videos locales y su metadato básico (ej. nombre, ubicación, estado), para facilitar navegación, selección y transferencia. |
| **General** | El módulo actúa como soporte del Controlador Multimedia y del Servicio LAN. Su entrada son solicitudes de consulta o actualización (ej. "siguiente video"), y su salida son rutas de archivos o descriptores multimedia que el CM utiliza para ejecutar la acción. |
| **Marco Técnico del Módulo** | * **Almacenamiento local:** directorio estructurado de archivos de video. * **Gestión de metadatos:** SQLite o JSON indexado para listar videos y estados. * **Interfaz de acceso:** API interna consultada por el Controlador Multimedia. |
| **Estructura General** | 1. Consultar lista de videos disponibles. 2. Retornar referencia al video solicitado (ej. siguiente, anterior). 3. Actualizar estado de reproducción o transferencia. 4. Proveer información al Servicio LAN en caso de envío remoto. |

# Diseño de Interfaces

## Modelo de Navegación

Diagrama general que muestra las interacciones (eventos o acciones) entre los elementos de interfaces usuarias del sistema. Ejemplos (nótese que en ambos modelos el punto de partida es claro):





## Prototipos de Interfaces Usuarias

[Repetir para las GUI pincipales del sistema]

### Prototipo [XX]

[Mostrar gráficamente el diseño de la interfaz (no funcional). El nombre referenciado debe ser coherente con el nombre correspondiente especificado en 3.1. Detallar los eventos que se pueden gatillar desde la interfaz, completando la siguiente tabla:]

|  |  |  |  |
| --- | --- | --- | --- |
| **Evento** | **Interacción** | **Acción** | **Objeto afectado** |
| **[Nombre del evento]** | [Descripción de acción que gatilla el evento (e.g apretar un botón, tocar pantalla...).] | [Descripción de la acción resultante del evento. (e.g. Se abrió ventana, se muestra otro elemento, mensaje)] | [Objeto sobre el cual se ejerce la acción. Ser coherente con los nombres especificados en 3.1 y en la visualización gráfica de la interfaz]. |

## 