

**Escuela Superior Politécnica del Litoral**

**Facultad de Ingeniería en Electricidad y Computación**

Implementación de un módulo de etiquetado de gestos y posturas para el sistema de retroalimentación automática de presentaciones RAP

**Proyecto Integrador**

Previo la obtención del Título de:

**Nombre de la titulación**

Ingeniera/o en Ciencias de la Computación

Presentado por:

Lizbeth Damaris Vergara Castro

Ricardo Mitridates Zevallos Mosquera

Guayaquil - Ecuador

Año: 2023

## **Agradecimientos**

Quiero agradecer a mis padres Leonor Castro y Klinton Vergara por apoyarme en cada paso de mi vida, a todas mis tías por ser un apoyo incondicional en mi educación, a mi prometido Brian Lara por su paciencia, respaldo y amor. ¡Y un saludo al cielo, gracias mamita marina lo logramos!

**Lizbeth Vergara**

### **Declaración expresa**

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Lizbeth Damaris Vergara Castro y Ricardo Mitridates Zevallos Mosquera damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”.



Lizbeth Damaris  
Vergara Castro



Ricardo Mitridates  
Zevallos Mosquera

## **Evaluadores**

---

**Miguel Realpe**

Profesor de la materia

---

**Federico Domínguez**

Profesor tutor

## Resumen

La retroalimentación para los modelos de inteligencia artificial es crucial para su entrenamiento, este proyecto aborda la integración de un módulo de etiquetado de gestos y posturas en el sistema de retroalimentación automática de presentaciones (RAP) de la Escuela Superior Politécnica de Litoral que se implementa en videos que necesitan retroalimentación. Presenta la importancia de la inteligencia artificial en la educación y la necesidad de integrar un módulo dentro del sistema RAP. Se plantea un problema de ineficiencia en el proceso de retroalimentación manual de videos de presentaciones y se justifica la implementación del módulo de etiquetado como solución. Los objetivos incluyen la implementación del módulo utilizando Django, la visualización de videos a etiquetar, la creación de etiquetas de gestos y posturas, y la generación de archivos CSV con etiquetas creadas. El desarrollo se centra en la adaptación del sistema de etiquetado existente y la creación de una interfaz a partir de dicho sistema. Los resultados mostraron un módulo funcional que simplificó el etiquetado manual, donde Los usuarios pueden crear y eliminar etiquetas en tiempo real, y generar archivos CSV con las etiquetas guardadas. La implementación del módulo de etiquetado manual mejora significativamente la eficiencia y calidad del sistema RAP, beneficiando a estudiantes, profesores y miembros del equipo de tecnología de la información. El proyecto logra sus objetivos y presenta un avance importante en la retroalimentación automática de presentaciones.

**Palabras Clave:** retroalimentación automática de presentaciones, etiquetado, *Django*, *CSV*, videos.

## Abstract

*Feedback for artificial intelligence models is crucial for their training. This project addresses the integration of a gesture and posture labeling module into the automatic feedback system for presentations (RAP) at the Escuela Superior Politécnica de Litoral, which is implemented in videos requiring feedback. It highlights the importance of artificial intelligence in education and the need to integrate a module into the RAP system. An inefficiency problem in the manual feedback process for presentation videos is identified, and the implementation of the labeling module is justified as a solution. The objectives include implementing the module using Django, displaying videos for labeling, creating gesture and posture labels, and generating CSV files with the created labels. The development focuses on adapting an existing labeling system and creating an interface based on it. The results demonstrate a functional module that simplifies manual labeling, allowing users to create and delete labels in real-time, and generating CSV files with saved labels. The implementation of the manual labeling module significantly improves the efficiency and quality of the RAP system, benefiting students, teachers, and IT team members. The project successfully achieves its goals and represents a significant advancement in automatic presentation feedback.*

**Keywords:** automatic feedback for presentations, labeling, Django, CSV, videos.

## Índice general

Resumen .....	I
Abstract.....	II
Índice general .....	III
Abreviaturas.....	V
Índice de figuras.....	VI
Índice de tablas.....	VIII
1.    Capítulo 1.....	1
1.1    Introducción .....	2
1.2    Descripción del problema.....	3
1.3    Justificación del problema.....	3
1.4    Objetivos.....	6
1.4.1    Objetivo general.....	6
1.4.2    Objetivos específicos .....	6
1.5    Marco teórico .....	6
2.    Capítulo 2.....	10
2.1    Metodología .....	11
2.1.1    Análisis .....	11
2.1.2    Prototipado .....	15
2.1.3    Diseño de la solución.....	19
3.    Capítulo 3.....	26
3.1    Resultados y análisis .....	27

3.1.1	Módulo de etiquetado. ....	27
3.1.2	Creación De Etiqueta. ....	28
3.1.3	Eliminación de etiqueta. ....	30
3.1.4	Guardado de etiquetas en archivo CSV. ....	31
3.2	Análisis de costos. ....	32
3.2.1	Análisis de requerimientos. ....	32
3.2.2	Prototipado. ....	33
3.2.3	Desarrollo de Código. ....	33
3.2.4	Total de costos. ....	33
3.3	Entregables. ....	34
4.	Capítulo 4. ....	35
4.1	Conclusiones y recomendaciones. ....	36
4.1.1	Conclusiones ....	36
4.1.2	Recomendaciones ....	37
	Referencias. ....	38
A.	Apéndice ....	39
	Requerimientos funcionales. ....	40
	Flujo de ventanas ....	41
	Documentación del proyecto ....	44

### **Abreviaturas**

ESPOL	Escuela Superior Politécnica del Litoral
CTI	Centro de Tecnología de la Información
CSV	Comma Separated Values
IA	Inteligencia Artificial
RAP	Retroalimentación Automática de Presentaciones
SSH	Secure Shell
JS	JavaScript

## Índice de figuras

Figura 2.1: Diagrama de casos de uso.....	14
Figura 2.2: Listado de resultados del RAP .....	15
Figura 2.3: Módulo de etiquetado.....	16
Figura 2.4: Selección de sección video a etiquetar .....	16
Figura 2.5: Selección de etiqueta .....	17
Figura 2.6: Descarga del CSV con las etiquetas correspondientes .....	17
Figura 2.7: Diagrama de bloques .....	19
Figura 2.8: Diagrama de secuencia .....	21
Figura 2.9: Diagrama de flujo .....	22
Figura 2.10: Diagrama de la base de datos modificada .....	23
Figura 2.11: Tabla Etiqueta añadida a la base de datos original.....	24
Figura 3.1: Acceso al módulo de etiquetado desde el RAP .....	27
Figura 3.2: Vista general del módulo de etiquetado .....	28
Figura 3.3: Creación de etiqueta .....	29
Figura 3.4: Creación de etiqueta con modificaciones pedidas por el cliente .....	29
Figura 3.5: Eliminación de etiquetas .....	30
Figura 3.6: Descarga de las etiquetas del video en formato CSV .....	31
Figura 3.7: Archivo CSV generado.....	32
Figura A.1: Requerimientos del proyecto .....	40
Figura A.2: Documento con flujo de ventanas parte 1 .....	41
Figura A.3: Documento con flujo de ventanas parte 2 .....	42
Figura A.4: Documento con flujo de ventanas parte 3 .....	43
Figura A.5: Documentación del proyecto parte 1 .....	44

Figura A.6: Documentación del proyecto parte 2 .....	45
Figura A.7: Documentación del proyecto parte 3 .....	46

**Índice de tablas**

Tabla 1.1: <i>Comparación de herramientas de anotación de video</i> .....	7
Tabla 3.1: <i>Costos de implementación</i> .....	34

## 1. Capítulo 1

## 1.1 Introducción

En la era actual, la inteligencia artificial (IA) ha adquirido una relevancia indiscutible en diversos ámbitos de nuestra vida cotidiana. Uno de los campos donde se destaca su aplicación es la educación, donde la IA se ha convertido en una herramienta valiosa para mejorar la retroalimentación y el aprendizaje de los estudiantes. En este contexto, surge la problemática en el sistema de retroalimentación automática de presentaciones (RAP), que busca retroalimentar precisa y efectiva mediante el análisis de videos de presentaciones de los estudiantes.

Sin embargo, a pesar de los avances en inteligencia artificial, el RAP no siempre logra retroalimentar correctamente las presentaciones, lo que implica un desafío en la mejora continua del modelo. Actualmente, cuando un profesor detecta un video que no está correctamente procesado por el RAP, se debe realizar un reporte que luego revisa y procesa manualmente un miembro del equipo encargado de etiquetar los gestos presentes en el video. Esta tarea adicional se lleva a cabo en un sistema separado del RAP, lo que implica un proceso lento y tedioso, generando ineficiencias en el sistema.

En este contexto, se plantea la necesidad de implementar un sistema de etiquetado de posturas como módulo integrado del RAP. Esto permitiría eliminar la etapa de envío manual de información al servidor de etiquetado de gestos, agilizando así el proceso de retroalimentación y reentrenamiento del modelo de inteligencia artificial. Al unificar estos dos sistemas en uno solo, se busca reducir el tiempo y la dificultad asociados al proceso de mejora continua del sistema de retroalimentación automática de presentaciones.

El objetivo de este proyecto es proponer e implementar una solución efectiva para integrar el sistema de etiquetado de posturas como módulo del RAP. Se analizarán diferentes enfoques y técnicas que permitan optimizar el procesamiento manual de los videos de presentaciones, asegurando una retroalimentación precisa y oportuna para los estudiantes.

Asimismo, se evaluará el impacto de esta integración en términos de eficiencia, calidad y experiencia del usuario.

## **1.2 Descripción del problema**

La problemática surge en el sistema RAP, que no siempre logra retroalimentar correctamente las presentaciones que dan los estudiantes, por lo que estos videos tienen que procesarse manualmente para reentrenar el modelo de inteligencia artificial que se emplea. Esta tarea no es sencilla porque hay un sistema completamente separado del RAP que sirve para etiquetar manualmente los gestos de los videos que no pudieron procesarse correctamente en el RAP.

Cuando un profesor detecta un video que no está correctamente procesado en el RAP, envía un reporte al centro de tecnologías de información (CTI), donde los miembros designados deben extraer el id, buscarlo en la base de datos del RAP, copiarlos y enviarlos al servidor de etiquetado de posturas. Una vez que los datos están en el servidor de etiquetado, se procesan los videos manualmente, añadiendo los gestos y posturas que se dan en el video. Finalmente, hay que copiar de nuevo los datos procesados manualmente y enviarlos al servidor donde se realiza el reentrenamiento del modelo.

Al existir estos tres sistemas por separado, todo este proceso conlleva mucho tiempo y dificultades de por medio.

El requerimiento de este proyecto abarca la implementación del sistema de etiquetado de posturas como módulo integrado del RAP, dejando fuera la parte de envío de información de los gestos manualmente procesados hacia el servidor donde se reentrena el modelo.

## **1.3 Justificación del problema**

El proyecto debe realizarse para abordar el problema existente en el sistema RAP, un proceso tedioso que consume mucho tiempo y recursos debido a que hay tres sistemas actuando por separado. La ejecución del proyecto traerá varios beneficios e impactos:

- Ahorro de tiempo y recursos: Al eliminar la necesidad de procesar manualmente los datos de los videos y enviarlos al servidor del sistema de etiquetado, se reducirá significativamente el tiempo y los recursos necesarios para el proceso, permitiendo que profesores, ingenieros y ayudantes que pertenecen al CTI se enfoquen en otras tareas importantes.
- Reducción tiempo de uso: Al integrar el sistema de etiquetado de posturas como parte del RAP, se eliminará la necesidad de que los profesores tengan que reportar los videos a los miembros del CTI y que estos últimos copien datos manualmente entre sistemas separados. Esto simplificará el proceso y reducirá el tiempo que los encargados en el sistema, lo que resultará en una mayor eficiencia y menos posibilidades de errores.
- Mejora en la retroalimentación automática: Al integrar el sistema de etiquetado de posturas en el RAP, se espera que las presentaciones de los estudiantes mal procesados por el RAP sean corregidas de manera más precisa y efectiva. Esto mejorará el modelo de inteligencia artificial empleado en el RAP, que a su vez afectará a la calidad de la retroalimentación recibida por los estudiantes y les ayudará a mejorar sus habilidades de presentación.

La ejecución del proyecto es altamente conveniente debido a los beneficios y mejoras que se obtendrán. El proyecto aborda directamente una problemática existente y propone una solución práctica para mejorar el sistema RAP. Al eliminar la necesidad de procesamiento manual de la transferencia de datos logrando una mayor eficiencia y precisión en la retroalimentación, lo cual es altamente conveniente para los miembros del CTI, profesores y estudiantes. Adicionalmente, la ejecución del proyecto tendrá varias implicaciones prácticas:

- Integración del sistema de etiquetado de posturas: El sistema de etiquetado de posturas se convertirá en un módulo integrado del RAP, lo que implicará cambios en la arquitectura y funcionamiento del sistema. Esto requerirá la implementación de nuevas

funcionalidades y la integración de algoritmos de etiquetado de posturas en el RAP existente.

- Eliminación del proceso de transferencia de datos: Al integrar el sistema de etiquetado de posturas, se eliminará la necesidad de procesar manualmente los datos de un servidor a otro.
- Mejora en la eficiencia: La eliminación de la necesidad de procesamiento manual de la transferencia de datos y la simplificación del flujo de trabajo reducirá los tiempos de procesamiento.

Finalmente, La ejecución del proyecto beneficiará a varios grupos:

- Estudiantes: Los estudiantes se beneficiarán al recibir una retroalimentación más precisa y efectiva sobre sus presentaciones. Esto les ayudará a mejorar sus habilidades de presentación y comprensión de gestos y posturas.
- Profesores: Los profesores se beneficiarán al tener un sistema de retroalimentación automática más eficiente y preciso. Esto les permitirá ahorrar tiempo a la hora de reportar los videos mal procesados.
- Miembros del CTI: Al eliminar la necesidad de procesamiento manual de transferencia de datos, los miembros del CTI tendrán más tiempo y recursos disponibles para trabajar en otras áreas del proyecto o en el desarrollo de nuevas funcionalidades.
- ESPOL: Se beneficiará de un sistema de retroalimentación automática más avanzado y eficiente. Esto mejorará la calidad de la enseñanza y la experiencia de aprendizaje de los estudiantes, lo que a su vez puede aumentar la reputación de la institución.

## **1.4 Objetivos**

### **1.4.1 *Objetivo general***

Implementar un módulo que replique las funcionalidades de etiquetamiento manual de gestos y posturas del sistema existente, usando la herramienta Django para lograr una integración exitosa con la plataforma RAP.

### **1.4.2 *Objetivos específicos***

1. Desarrollar una interfaz en el framework Django que facilite a los usuarios acceder al módulo de etiquetamiento en la página principal del sistema RAP.
2. Establecer una funcionalidad que permita la visualización de una lista de vídeos que presenten errores de etiquetamiento automático.
3. Elaborar una interfaz que permita al usuario etiquetar gestos y posturas correspondientes del video seleccionado.
4. Implementar un generador de archivos CSV que contenga la información del etiquetado manual con las líneas de tiempo correspondientes del video seleccionado.

## **1.5 Marco teórico**

Este proyecto se desarrollará en Django para integrarse con el RAP. El principal reto del módulo a desarrollar requiere resolver que herramientas de etiquetamiento de videos se va a emplear. Para este fin se ha investigado distintos proyectos y herramientas existentes que dispongan de dicha funcionalidad.

En [1] se revisan las herramientas de anotación de videos existentes. Los autores plantean la usabilidad y características de las distintas herramientas, destacando las ventajas de cada una y comparándolas en la tabla 1.1.

**Tabla 1.1***Comparación de herramientas de anotación de video*

Parameters vs Tool	iVAT	ViTBAT	MViPER-GT	BeaverDam	VATIC
Platforms/Programming Language used	C,C++ libraries/Qt library/Open Source Computer Vision Library	MATLAB/Computer Vision Toolbox	Java Open Source Project/ ViPER API	Cloud based/ Python GUI/Django	Amazon Mechanical Turk
Boundary/ Shape of the Object	Rectangle/Eclipse/ Polygon	Rectangle/Eclipse/ Polygon	Rectangle/Eclipse/ Polygon	Rectangle	Rectangle
Targetting	Individual	Individual, Group	Individual, Group	Individual	Individual
Interface	Three modules: semi, semi-automatic and automatic detection using various detection algorithms	Improved behaviour tracking of crowds	Improved tracking for enhanced real time video analysis	Cloud interface removes the problems of command line interface of ViPER	Crowd sourced Video annotation
Customisability	Automated Tracking with the option of manual assist	Can target individual human behaviour	Capable of developing automatic annotations at the semantic level	User friendly GUI and easy to install	Suitable to work in different application domains
Machine Learning Algorithms used, if any	Supervised Object Detector Algorithms	No	Semi-Supervised Learning	No	No

Nota. Tabla extraída de "Video annotation tools: A Review" [1].

En la tabla 1.1 se muestran 6 tecnologías de anotación de videos muy utilizadas en la actualidad, los autores comparan estas herramientas en términos de las plataformas en las que funcionan, las formas de objetos que pueden procesar, los objetivos que permiten anotar, el tipo de interfaz que ofrecen y si utilizan aprendizaje automático. Adicionalmente, se describen los distintos lenguajes empleados en el desarrollo de las herramientas analizadas, entre las que están C, C++, QT, MATLAB, Java, Python y Django. Teniendo esto en cuenta, se investigaron las que empleaban Python o Django para su desarrollo, las cuales fueron VATIC y BeaverDam.

VATIC es una plataforma de código abierto desarrollado en [2], lo que permite a investigadores contribuir y mejorar el sistema de etiquetado de videos para usarse en un proceso posterior de análisis y comprensión. El sistema destaca por su enfoque en su

simplicidad y flexibilidad, basado en tecnologías web, desarrollaron además una interfaz intuitiva que el usuario pueda realizar anotaciones en videos de manera eficiente.

Sus principales características son la capacidad de establecer tipos de anotaciones o etiquetas, regiones de interés, la posibilidad de asignar diferentes niveles de acceso y permisos a los usuarios. Además, luego del etiquetado de video la herramienta tenía acceso de revisión y validación de las anotaciones, lo que permitía garantizar la calidad de los datos anotados.

Por otro lado, en [3] se presenta BeaverDam, una herramienta de anotación de videos basada en python3, Django y HTML5 video. BeaverDam tiene el propósito de mejorar las limitaciones de las herramientas existentes en el campo. Su objetivo es mejorar la experiencia del usuario y permitir pruebas e iteraciones rápidas a través de una interfaz simplificada y un enfoque en la facilidad de instalación y configuración.

A diferencia de VATIC, BeaverDam ofrece una interfaz web fácil de usar para agregar y ver videos y trabajos, eliminando la necesidad de interacciones complejas en la línea de comandos. Además, proporciona una interfaz de línea de comandos en Python respaldada por Django para cargar videos en grandes cantidades y realizar otras tareas de manera programática. La herramienta enfatiza la modularidad y flexibilidad, lo que permite a los usuarios modificarla y personalizarla para adaptarse a sus necesidades específicas.

Explorando los distintos proyectos en GitHub relacionados con anotación o etiquetamiento de videos se encontraron dos proyectos desarrollados en Python que poseían un Frontend y Backend sencillo y simple. Uno de ellos es “multimedia annotator” [4] desarrollado por Dimitris Sgouropoulos, el cual posee módulos para etiquetado de audio y video para anotar fácilmente los videos por frames o por grupos formando áreas rectangulares. El otro proyecto, llamado “Bilkent Video Annotation Tool” [5] desarrollado por Selim Firat Yilmaz, está enfocado en ayudar en la anotación de posiciones en videos para su posterior procesado. También se analizó una herramienta desarrollada en

JavaScript llamada “videojs annotation comments” [6], la cual es un plugin perteneciente al reproductor de video web de código abierto video.js [7]. Este plugin permite a video.js la selección de secciones de video de una manera intuitiva, junto con un etiquetado de estas secciones eficiente y sencillo.

Respecto a que herramientas emplear a la hora de la generación de CSV, Django posee dos opciones. Una de ellas es emplear la librería “Python CSV”, haciendo que la creación de CSV actúe en objetos similares a archivos y los objetos HTTP Response pertenecientes a Django sean objetos similares a archivos. La otra opción es usar el sistema de plantillas de Django, el cual actúa a un nivel más bajo que la librería “Python CSV” [8].

## **2. Capítulo 2**

## 2.1 Metodología

### 2.1.1 Análisis

**Requerimientos.** El proyecto de etiquetado de posturas contará con los siguientes requerimientos:

- El módulo de etiquetamiento manual de posturas debe ser implementado utilizando el framework Django.
- El módulo debe ser compatible e integrable con el sistema RAP.
- El módulo debe proporcionar opciones para etiquetar gestos y posturas del video seleccionado mediante alguna herramienta de anotación de video.
- La herramienta de anotación de video debe permitir seleccionar una sección de tiempo del video correspondiente y asignarle un gesto o postura.
- La herramienta de anotación de video debe permitir guardar, modificar y eliminar las distintas anotaciones.
- El sistema debe generar archivos CSV que contengan la información del etiquetado manual.
- Los archivos CSV deben incluir las líneas de tiempo correspondientes a los gestos y posturas etiquetados del video seleccionado.

**Alcance y limitaciones.** La solución se limita a recibir videos proporcionados por el módulo resultados, el cual el CTI (Centro de Tecnología de la Información) se encargará modificar para que este módulo pueda mostrar solo los videos que necesiten ser procesados manualmente. Cuando se acceda al módulo de etiquetado manual mediante el módulo de resultados, se emplearán los datos asociados a el resultado correspondiente para encontrar el video guardado en el servidor del rap. Una vez se obtenga el video, se procesará manualmente el video añadiendo los gestos que se crean necesarios y se guardarán para la generación del CSV.

La selección de los videos que serán etiquetados no estará a cargo de la integración propuesta. Será responsabilidad del equipo de CTI proporcionar al sistema los videos que serán etiquetados. La solución se enfoca en la integración del módulo de etiquetamiento y el proceso de etiquetado en sí, pero no en la selección de los videos que serán editados.

Debido a la limitación mencionada anteriormente, la solución no ofrecerá opciones para elegir qué videos serán etiquetados. El equipo de CTI será responsable de marcar los videos que tuvieron una mala retroalimentación por parte del sistema el sistema solo realizara un filtro donde solo los videos que tengan cierta marca serán proporcionados al sistema para su etiquetado.

Los videos disponibles en el nuevo módulo de etiquetamiento estarán sujetos a la disponibilidad y provisión por parte del equipo de CTI. No se garantiza que todos los videos existentes en el sistema RAP estarán disponibles en el módulo de etiquetamiento.

Existe una limitación adicional en la solución propuesta, ya que el sistema RAP no estará directamente conectado con el modelo de IA utilizado para la evaluación de los datos etiquetados. En lugar de ello, el sistema RAP generará un archivo CSV que contendrá las etiquetas y la información correspondiente, con este archivo CSV el equipo de CTI deberá enviarse manualmente al modelo de IA para la evaluación adecuada.

Es importante tener en cuenta esta limitación, ya que implica una intervención manual por parte del equipo de CTI para transferir los datos etiquetados al modelo de IA. La solución no incluye una integración directa entre el sistema RAP y el modelo de IA para la transferencia automática de los resultados del etiquetado.

**Riesgos y beneficios.** La solución propuesta integra el sistema de etiquetamiento en el sistema RAP de ESPOL, lo que significa que ahora solo necesitarán utilizar un solo sistema para el etiquetamiento manual. Simplifica y agiliza el proceso de etiquetado, ahorrando tiempo y esfuerzo al equipo de CTI. Mejorando la calidad y tiempo del trabajo al tener un sistema unificado de etiquetamiento en el sistema RAP, se espera que la calidad del trabajo del equipo

de CTI mejore significativamente. Con una herramienta más efectiva y rápida, podrán realizar la retroalimentación de manera más eficiente, lo que permitirá una mayor calidad de los datos etiquetados y reducir el tiempo necesario para el proceso.

También permitirá una retroalimentación más efectiva y rápida para las personas que conforman ESPOL. Al tener un sistema de etiquetamiento integrado, podrán recibir comentarios y correcciones de manera oportuna, lo que facilitará su aprendizaje y desarrollo en el ámbito de etiquetamiento manual.

Es importante reconocer y abordar los posibles riesgos y desafíos asociados con la implementación de la solución propuesta. Debido a que el sistema de etiquetación lleva ya un tiempo existiendo de manera independiente al sistema RAP Puede haber resistencia al cambio por parte del equipo de CTI al adoptar el nuevo sistema de etiquetamiento integrado. Es importante gestionar adecuadamente este riesgo mediante la capacitación y la comunicación efectiva para garantizar una transición suave y exitosa.

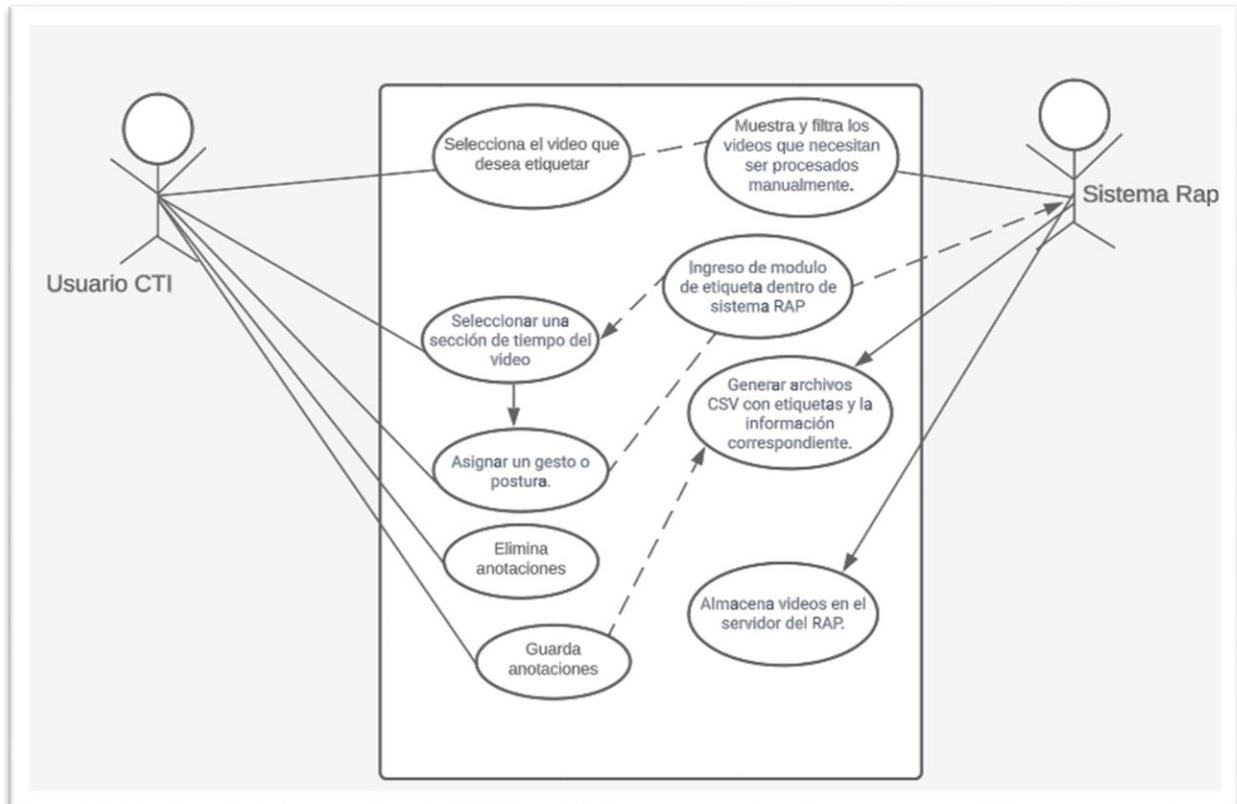
Podrían surgir errores en el sistema integrado que pueden afectar la precisión y confiabilidad de los resultados de etiquetado manual, esto mientras el equipo de CTI se acostumbre al nuevo flujo de trabajo y adquieran habilidades específicas para utilizar eficazmente el sistema de etiquetamiento integrado.

**Usuarios.** Los usuarios de esta solución son solo los miembros designados del CTI para esta tarea, por lo que estos usuarios tendrán permisos especiales para acceder al módulo de etiquetado manual.

## Casos de uso

**Figura 2.1**

Diagrama de casos de uso



Nota. Muestra cómo se relacionan los actores del sistema RAP y sus casos de uso. Los actores son el usuario CTI, usuario que interactúa con la plataforma y Sistema RAP, representa el sistema de etiquetado de videos.

El sistema RAP filtra los videos que necesitan ser etiquetados manualmente y los muestra, la relación depende de este paso por el sistema RAP, el usuario selecciona el video que desea etiquetar.

Cuando el usuario selecciona el video que desea etiquetar manualmente accede al módulo de etiquetamiento, la relación depende de la selección de video del usuario.

Para etiquetar el video, el usuario selecciona la sección de video que desea etiquetar, luego se debe asignar un gesto o postura y el usuario podría eliminar alguna anotación si se requiere.

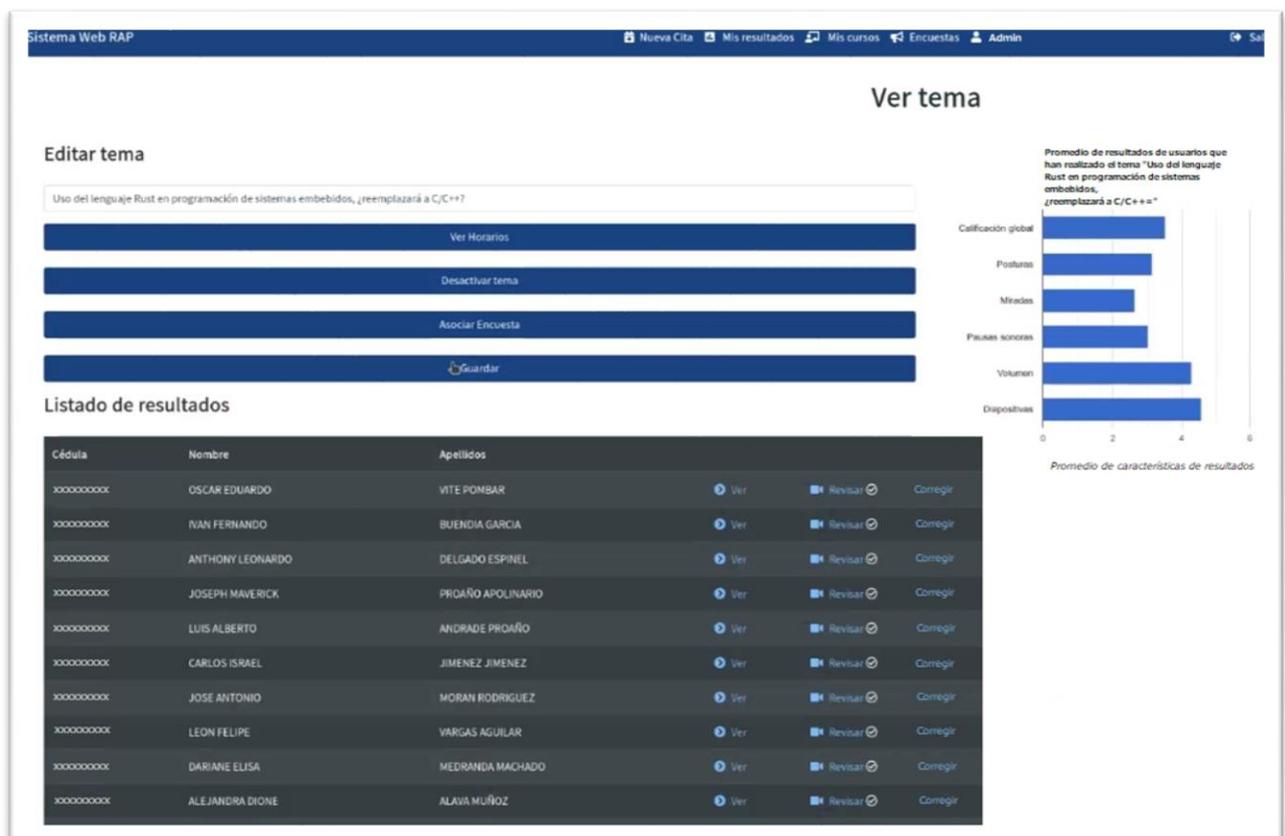
Cuando el usuario ha terminado de etiquetar el video, guarda las anotaciones esta es la relación con el caso de uso, luego el sistema RAP genera un archivo CSV con la información de las etiquetas correspondiente.

### 2.1.2 Prototipado

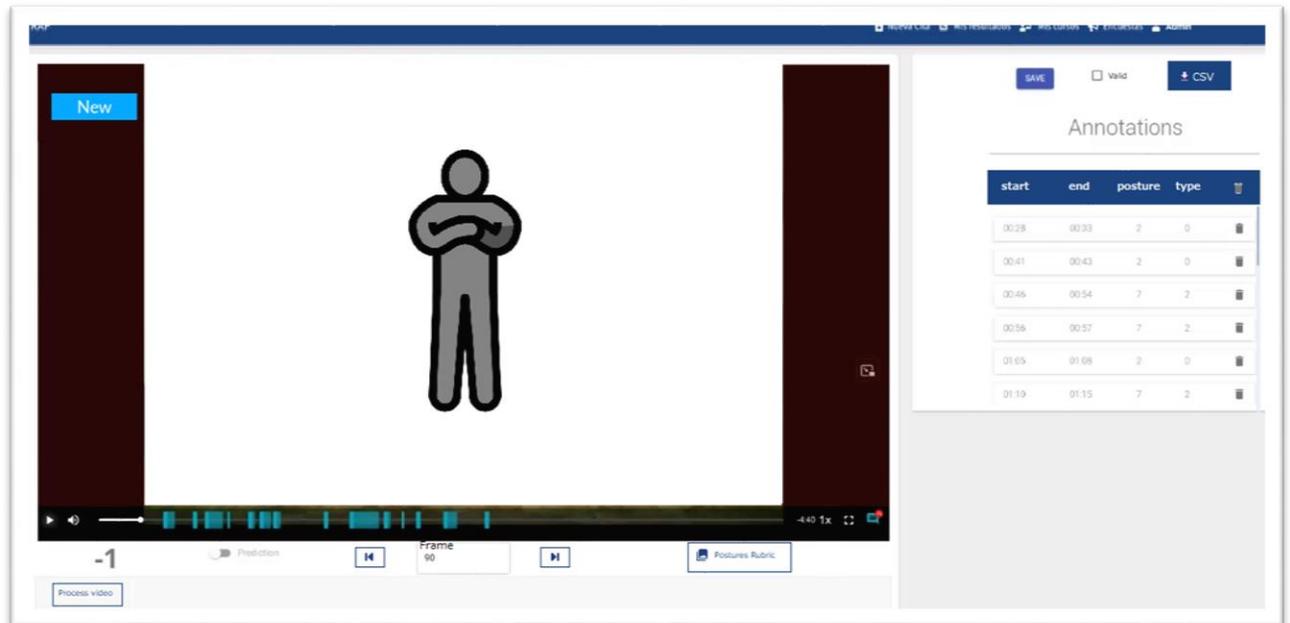
A continuación, se muestran las ventanas principales del prototipo creado [9] que conectan el RAP con el módulo de etiquetado.

**Figura 2.2**

*Listado de resultados del RAP*



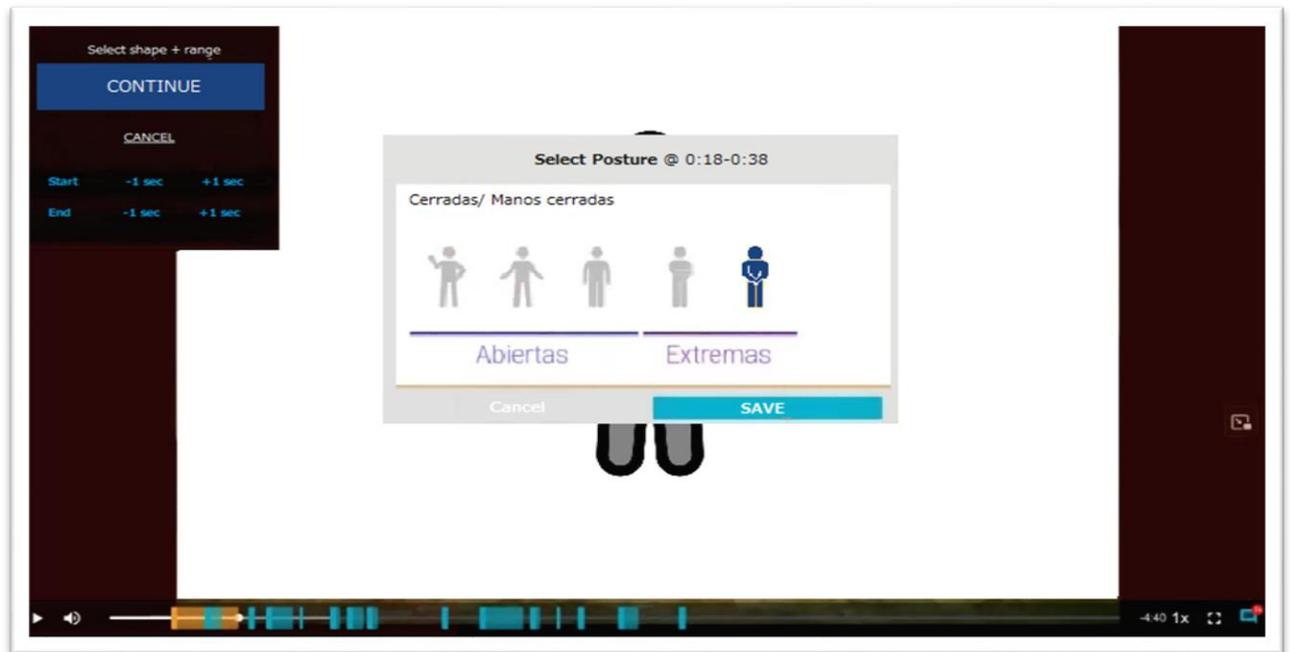
Nota. Primera ventana en el flujo.

**Figura 2.3***Módulo de etiquetado*

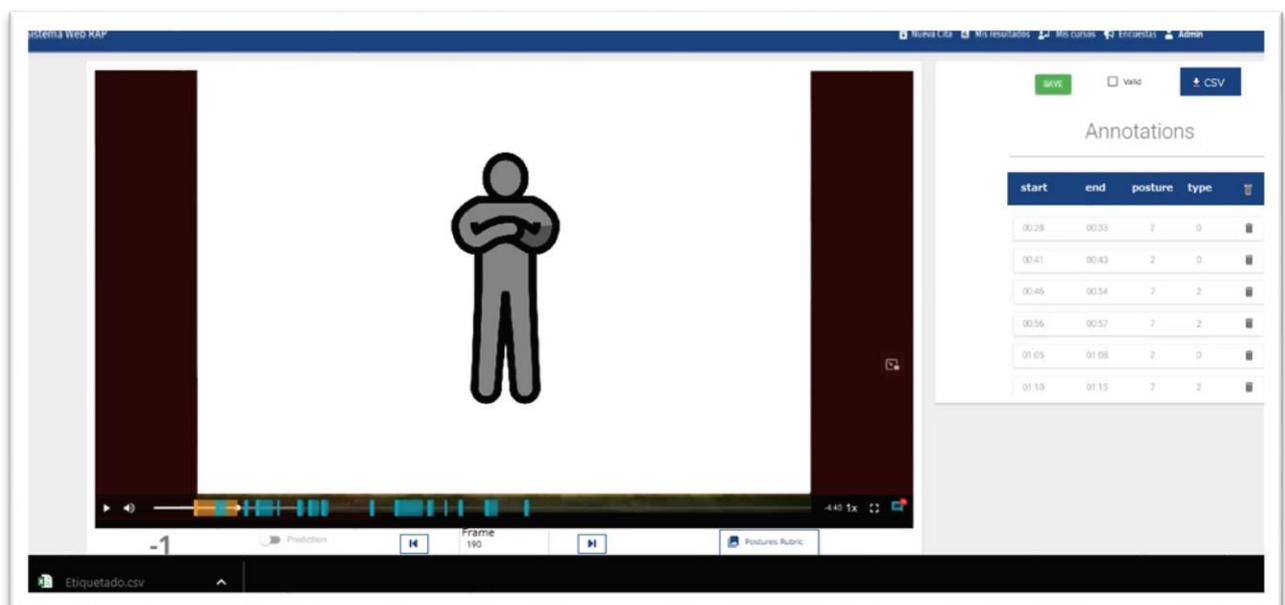
Nota. Segunda ventana en el flujo.

**Figura 2.4***Selección de sección video a etiquetar*

Nota. Tercera ventana en el flujo.

**Figura 2.5***Selección de etiqueta*

Nota. Cuarta ventana en el flujo.

**Figura 2.6***Descarga del CSV con las etiquetas correspondientes*

Nota. Quinta ventana en el flujo.

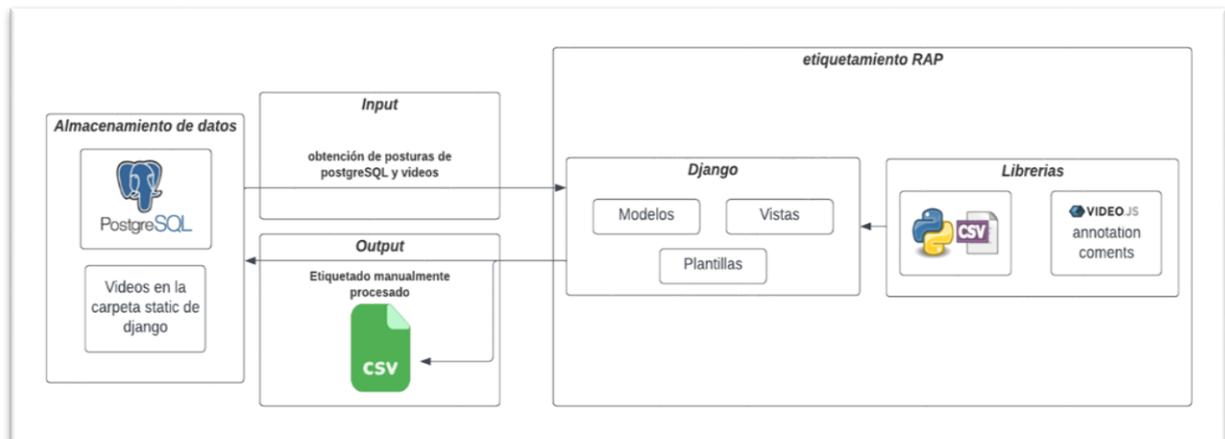
El flujo de ventanas comienza desde la página de resultados del RAP, la cual presenta las diferentes retroalimentaciones de los estudiantes. Los resultados que permiten acceder al módulo de etiquetado de posturas son los que están marcados para revisión manual. Una vez se decide que video procesar manualmente, se le da al botón procesar y se accederá a la ventana de etiquetado manual. Esta ventana permite crear una nueva etiqueta, que seleccionará una sección de tiempo del video junto a una postura concreta, para luego guardarla y poder generar un archivo CSV con los datos resultantes [9].

Se probó el prototipo con usuarios de CTI con el objetivo de obtener comentarios y observaciones sobre su funcionalidad y usabilidad. Inicialmente, se presentó a los usuarios el primer prototipo propuesto. Sin embargo, durante las pruebas, se destacó que el prototipo tenía un aspecto visual diferente al del sistema RAP en el que estaría integrado, lo que podría generar confusiones a los usuarios. Se realizó ajustes en el color del módulo y el prototipo final se adaptó al mismo esquema de colores del sistema RAP. Esta modificación permitió la identificación visual del módulo dentro del sistema, evitando que el usuario pueda pensar que esta el otro sistema externo al RAP asegurando una experiencia de usuario más cohesionada. Cuando se presentó esta mejora, se realizó pruebas adicionales con los usuarios de CTI, quienes destacaron que el flujo del prototipo era intuitivo y fácil de seguir. La comprensión de las funcionalidades fue clara, lo que resultó en un prototipo eficiente y sin problemas de navegación. Estos aprendizajes obtenidos a través de las pruebas con usuarios contribuyeron en el diseño de la solución final, asegurando que se ajustara adecuadamente a las necesidades y expectativas de los usuarios, ofreciendo una experiencia de usuario más satisfactoria y efectiva en el sistema RAP. En siguientes iteraciones se realizará un cambio en la ventana mostrada en la figura 2.4 para añadir miradas, lo cual se verá reflejado en la implementación del prototipo.

### 2.1.3 Diseño de la solución

**Figura 2.7**

Diagrama de bloques



Nota. Muestra la estructura del proyecto y las librerías que se van a añadir fuera de las ya existentes en el RAP.

**Arquitectura.** Para el análisis de las funcionalidades y diseño del sistema de etiquetado de posturas, se empleó los datos del repositorio en donde está guardado. Este sistema emplea React para el Frontend y JavaScript junto con MongoDB para el Backend. El Frontend posee un *plugin* de video.js que se emplea para poder realizar las anotaciones manuales en determinadas secciones de video, lo que permite clasificar los gestos que se den en ese periodo de tiempo. Se emplea REACT-CSV para la funcionalidad de generar CSV de los gestos anotados manualmente.

Para la creación del módulo que replique estas funcionalidades, se analizaron y usaron las tecnologías que emplea el RAP. Este sistema emplea Django en el Backend junto a una base de datos PostgreSQL. El Frontend emplea el sistema de plantillas de Django para generar paginas HTML dinámicas, por lo que usa una arquitectura modelo vista plantilla, una variante del modelo vista controlador.

Entre todas las herramientas disponibles dentro de la anotación de videos, se optó por *Video.js*. En cuanto a la confiabilidad, posee actualizaciones constantes, mucha

documentación en línea y más de setecientos mil sitios web usándolo actualmente. Sobre la compatibilidad, aunque no se especifique que funciona en Django, es posible emplearlo con éxito debido a que Django permite el uso de archivos JS. Al ser la herramienta del módulo del sistema actual del etiquetado, ya existe una base para empezar a desarrollarlo. Respecto a la generación del CSV, se optó por Python CSV debido a la facilidad de uso, ya que las dos herramientas son confiables y compatibles con Django.

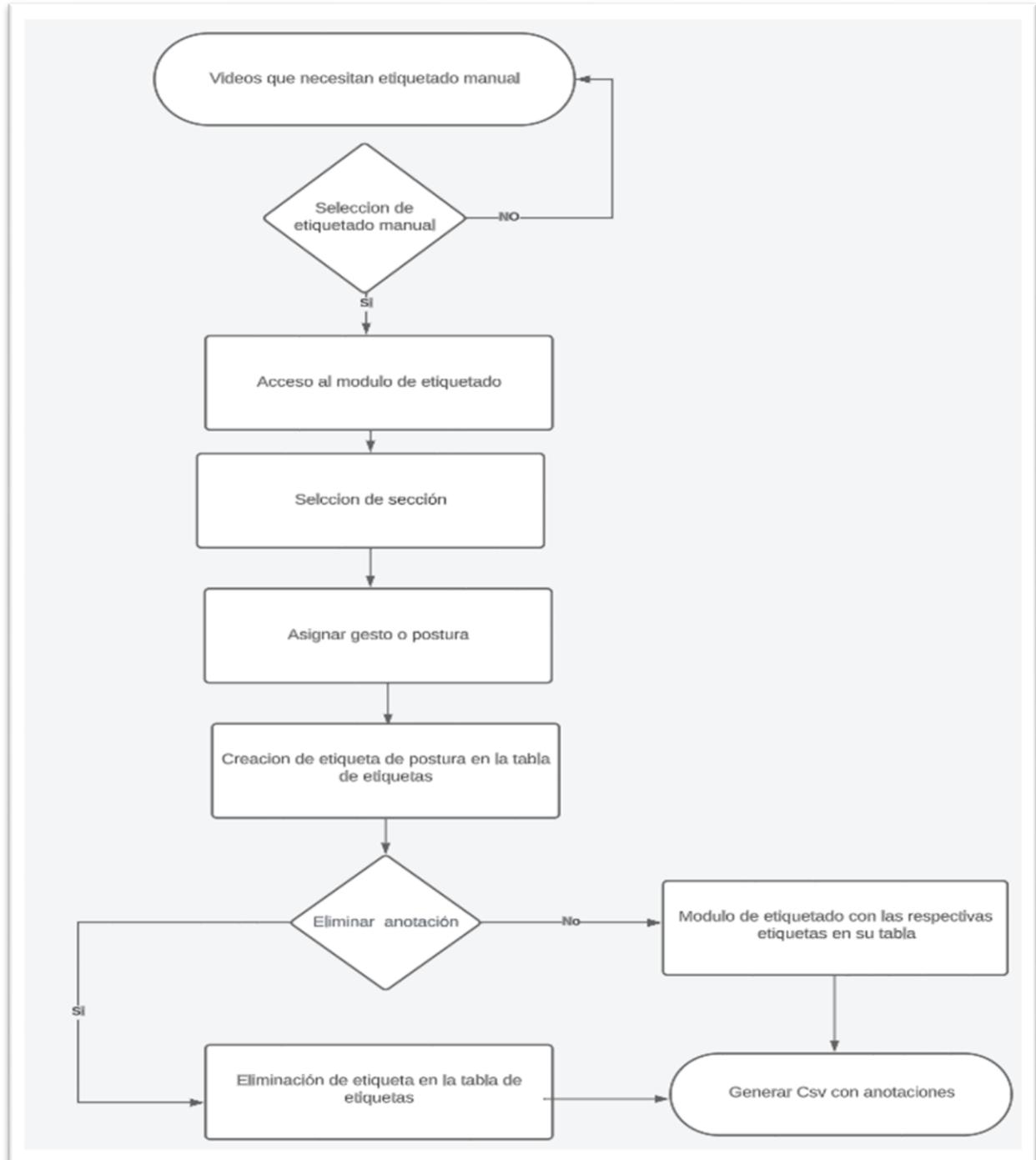
Para el nuevo sistema de etiquetado, se decidió emplear el código existente en una vista de un módulo llamado resultados, la cual posee el código para obtener los videos de un servidor mediante SSH. Una vez se haya obtenido el video y los datos asociados a este, *Video.js Annotation Comments* permitirá crear y borrar etiquetas del video, y *pythonCSV* generará las etiquetas en el formato deseado.

**Figura 2.8***Diagrama de secuencia*

Nota. Aquí se muestran las funciones principales del módulo.

**Relación de arquitectura del Frontend con el Diagrama de secuencia.** El Frontend del sistema es hecho en React con el plugin Video.js lo que permite a los usuarios de CTI realizar anotaciones manuales en el video seleccionado respetando la interacción del usuario con la interfaz.

**Relación de arquitectura del Backend con el Diagrama de secuencia.** El Backend del sistema utiliza JavaScript junto con MongoDB como base de datos que ayuda a la funcionalidad del sistema `videosConEtiquetadoManual()` que permite mostrar y seleccionar los videos que necesitan etiquetado manual.

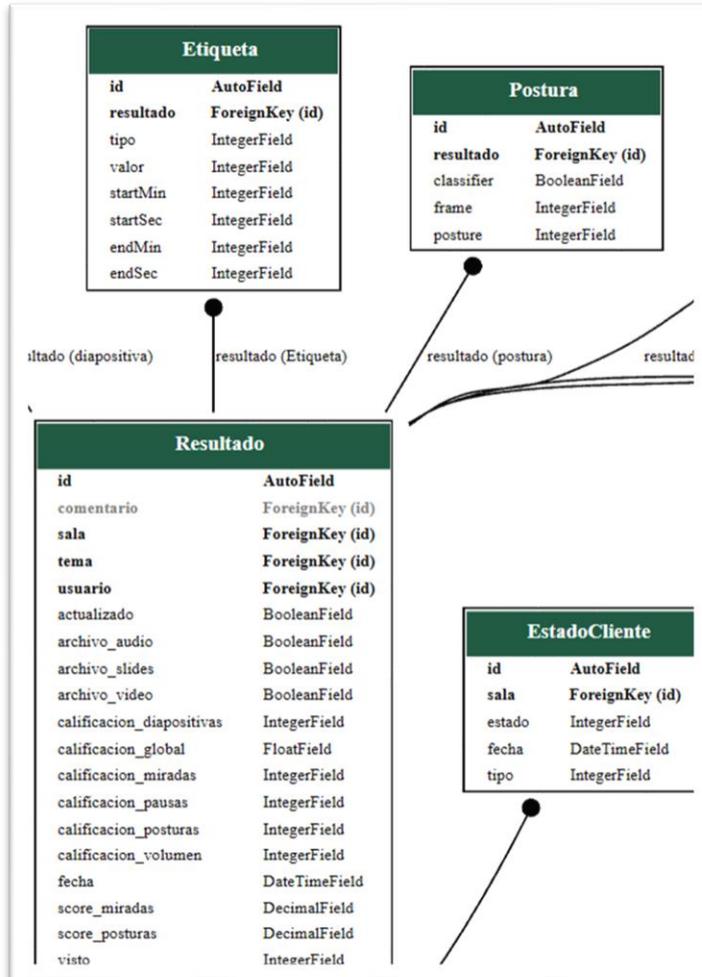
**Figura 2.9***Diagrama de flujo*

Nota. Diagrama que explica el flujo de un usuario para llegar a generar un CSV.



Figura 2.11

Tabla Etiqueta añadida a la base de datos original



Nota. La tabla añadida es Etiqueta.

En cuanto a la base de datos, se hizo una modificación a la existente, añadiendo la tabla Etiqueta. La tabla posturas posee distintas posturas asociadas a distintos frames de un video, pero eso son los resultados proporcionados por el modelo de inteligencia artificial. Para poder etiquetar un video mal procesado por el modelo de inteligencia artificial, no se tomaron en cuenta las posturas con sus frames asociados, sino que se decidió que el etiquetado fuese desde cero tomando en cuenta los minutos y segundos del video en cuestión. Esto se hizo debido a que el plugin para anotaciones no dispone de anotaciones tan precisas como frames,

pero los segundos fueron suficiente para poder retroalimentar al modelo de inteligencia artificial.

### **3. Capítulo 3**

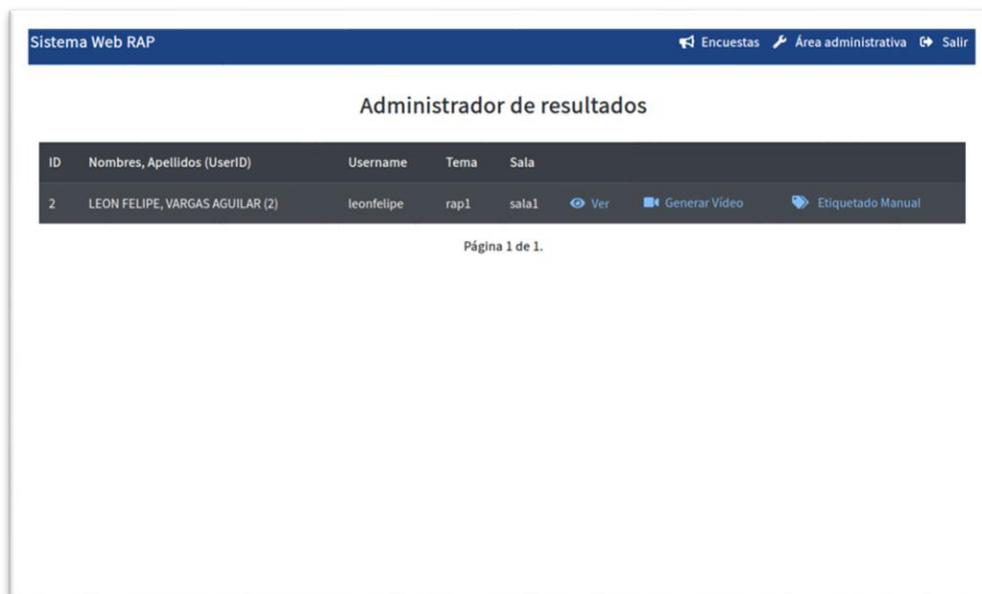
### 3.1 Resultados y análisis

#### 3.1.1 Módulo de etiquetado

Durante las pruebas del módulo de etiquetado, ciertos usuarios con permisos para retroalimentar manualmente videos en el RAP accedieron al sistema, ya que se despliega una lista de videos que necesitan el etiquetamiento manual, estos permisos de acceso los otorga personal del CTI. El usuario escoge un video y accede al módulo de etiquetado, este flujo fue probado por usuarios con el objetivo de que interactúen con la interfaz, dentro de la interfaz se realiza todo el proceso de etiquetado tales como, creación de etiqueta, eliminación de etiqueta y guardado de etiquetas en archivo CSV. Se recopiló información relevante para hacer mejoras, como la asignación de etiquetas sin restricciones, ya que se estableció un número limitado de etiqueta por fracciones de segundo, pero esto se retiró porque los usuarios necesitaban poner las etiquetas necesarias.

#### Figura 3.1

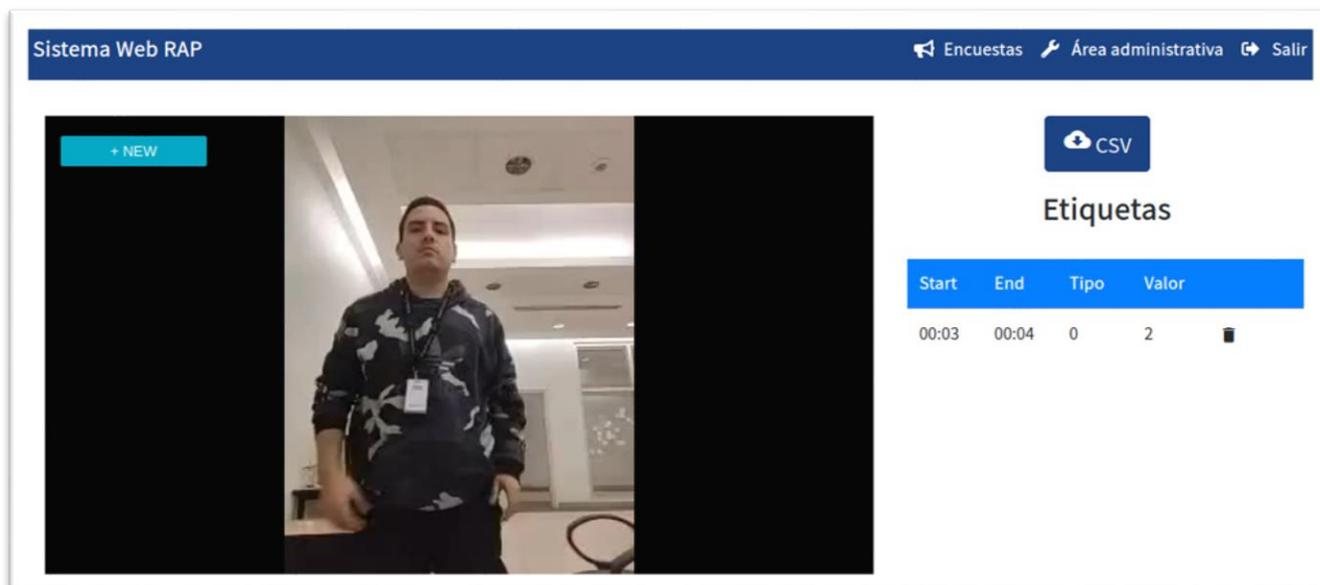
*Acceso al módulo de etiquetado desde el RAP*



Nota. Se accede desde el botón Etiquetado Manual.

**Figura 3.2**

Vista general del módulo de etiquetado



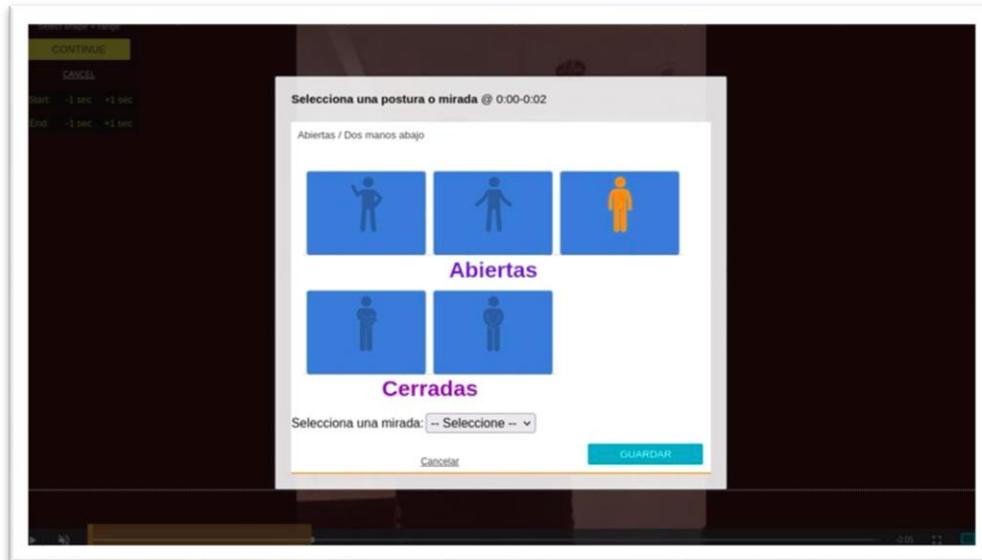
Nota. Esto es lo que se muestra al entrar al módulo de etiquetado.

### 3.1.2 Creación De Etiqueta

Se realizaron modificaciones y mejoras en el plugin *Video.js Annotation Comments*, estas modificaciones permitieron la creación de etiquetas, facilitando la integración de la funcionalidad en el módulo de etiquetado, las etiquetas se crean en postura con selección de recuadro o la selección de miradas en el cual se aplicó el evento *OnClick*, con una lista desplegable, que son guardadas en la tabla de etiquetas con la implementación de un *EventListener* específico que se activa en el momento que un usuario creaba una etiqueta, y se ve reflejado en la tabla de etiquetas garantizando la generación de la misma en tiempo real. En las pruebas realizadas con usuarios quedó evidenciada la facilidad de etiquetado que posee la interfaz, sin embargo, se realizó una modificación de color en la selección de las etiquetas de posturas, inicialmente se planteó en ponerlo de color azul pero finalmente se cambió por blanco.

**Figura 3.3**

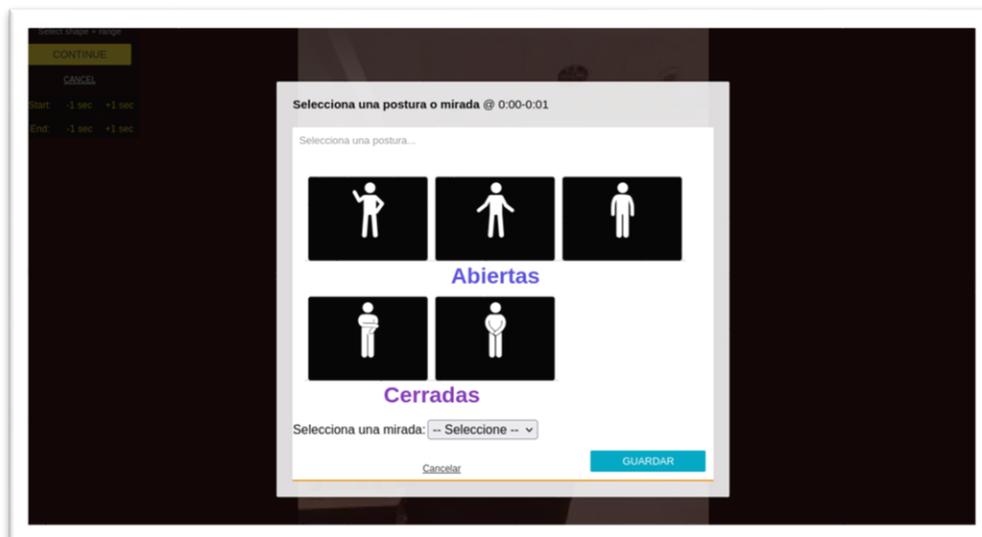
*Creación de etiqueta*



Nota. Esta es la creación de etiqueta inicial.

**Figura 3.4**

*Creación de etiqueta con modificaciones pedidas por el cliente*



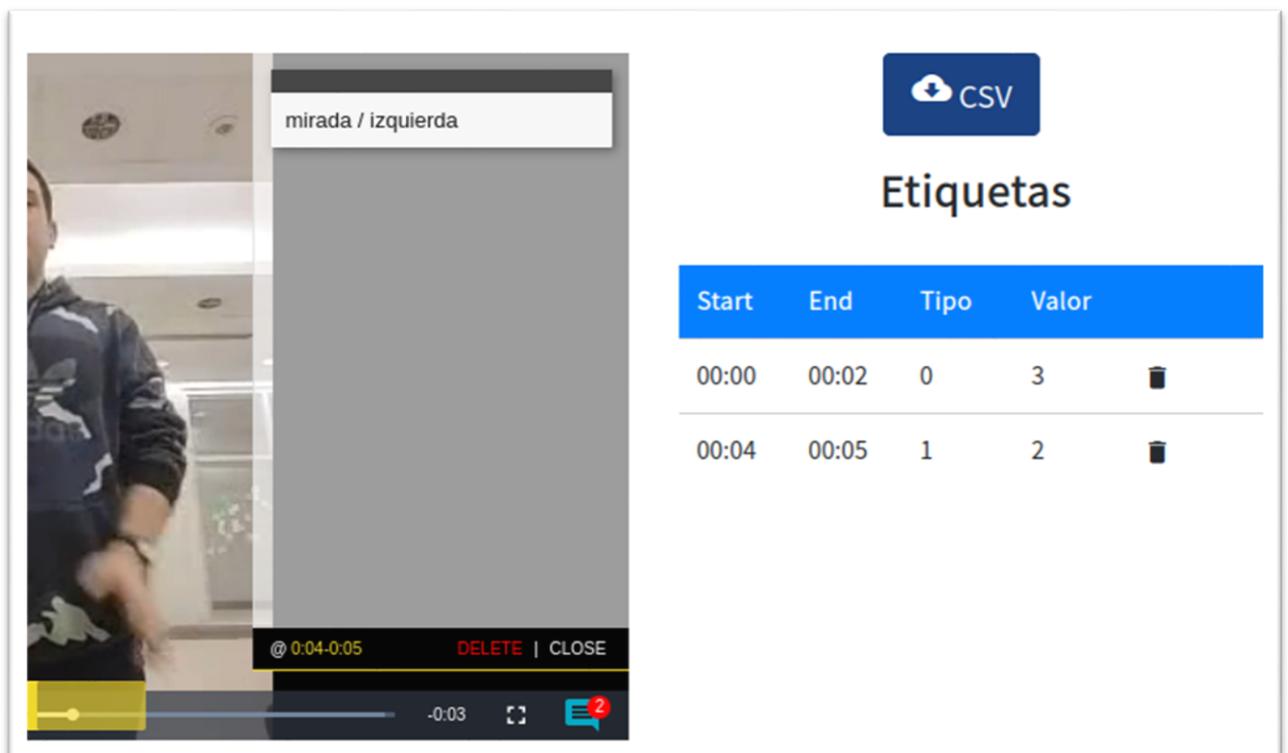
Nota. El cliente mencionó que los colores de los iconos no hacían buen contraste, así que se cambió por negro/blanco.

### 3.1.3 Eliminación de etiqueta

La eliminación de etiquetas fue esencial para el sistema de etiquetado, el plugin Video.js cuenta con funciones y eventos necesarios para realizar esta operación de manera eficiente y efectiva. El proceso de eliminación de etiquetas se realizó usando funciones proporcionadas por el plugin. Específicamente, la función `plugin.fire('destroyAnnotation', { id: id });`, además, se hizo uso del evento `plugin.registerListener('annotationDeleted', (event) => {})`; para gestionar las acciones que ocurren después de que una etiqueta es eliminada. La implementación de estas funciones y eventos resultó en una interacción fluida para la eliminación de etiquetas. Los usuarios del sistema pudieron interactuar con estas funcionalidades de manera intuitiva.

#### Figura 3.5

Eliminación de etiquetas



The screenshot shows a video player interface. On the left, a video is playing with a player control bar at the bottom showing a timestamp of 0:04-0:05 and a 'DELETE | CLOSE' button. On the right, there is a panel titled 'Etiquetas' (Annotations) with a 'CSV' button at the top. Below the title is a table with the following data:

Start	End	Tipo	Valor	
00:00	00:02	0	3	
00:04	00:05	1	2	

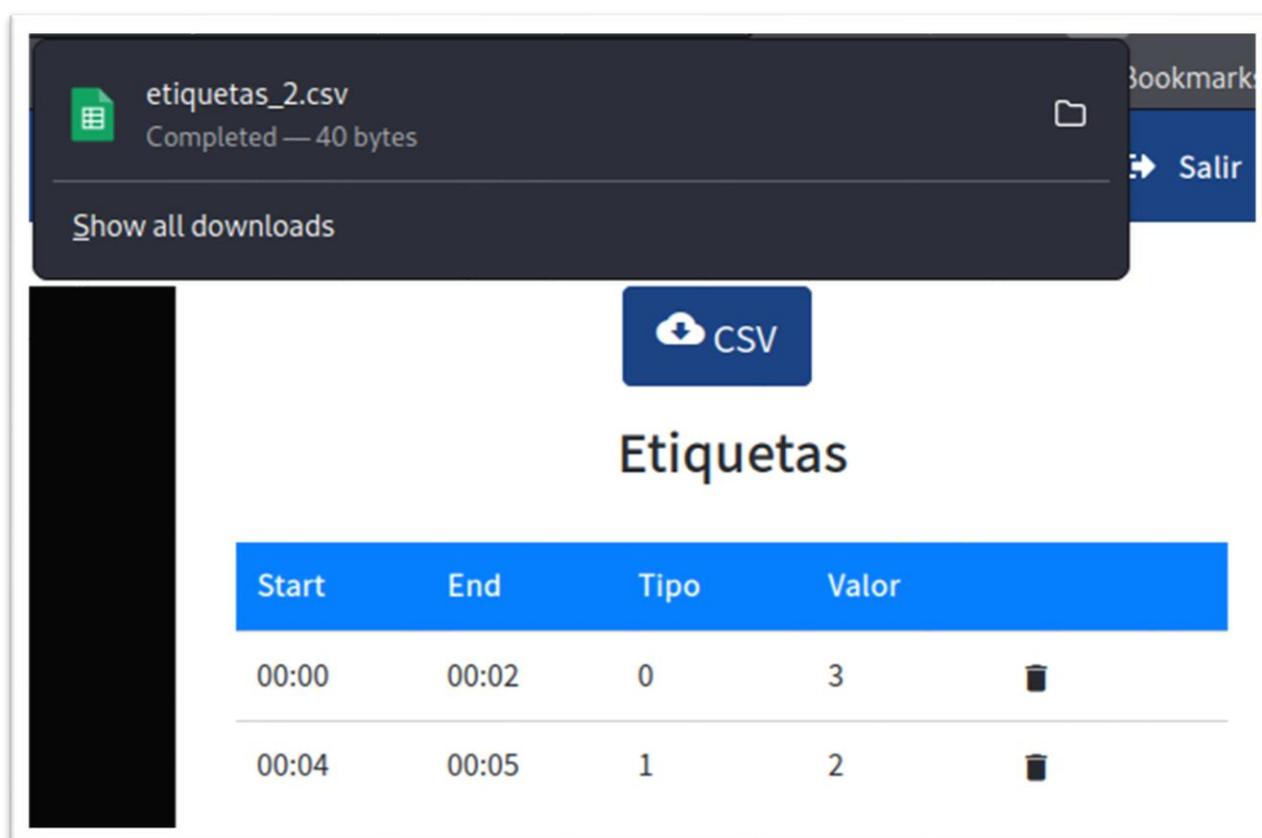
Nota. La etiqueta se puede borrar desde el botón rojo de *delete* dentro del plugin o con el icono de papelera en la tabla etiquetas de la derecha.

### 3.1.4 Guardado de etiquetas en archivo CSV

Se evaluó el proceso de guardado de etiquetas mediante la generación de un archivo CSV al concluir el etiquetado manual de los videos. Este paso se fue puesto a prueba cuando los usuarios optaban por concluir el etiquetado y guardar el CSV. Se compararon las etiquetas que habían sido creadas en la interfaz con la información de etiquetas guardadas en el archivo generado. Además, se revisó la estructura del archivo según lo establecido y se destacó la efectividad del sistema, ya que el archivo generado era lo que el usuario necesitaba, confirmando que no era necesario modificar el ordenamiento del mismo.

**Figura 3.6**

*Descarga de las etiquetas del video en formato CSV*



Nota. En el nombre del archivo se guarda el id de resultado, para saber a qué video pertenece el archivo generado.

**Figura 3.7**

Archivo CSV generado

	A	B	C	D	E	F	G	H	I
1	Start	End	Tipo	Valor					
2	0	2	0	3					
3	4	5	1	2					
4									
5									
6									
7									
8									
9									
10									
11									

Nota. Las columnas pertenecientes a *Start* y *End* proporcionan el tiempo en segundos totales.

### 3.2 Análisis de costos

Para analizar el costo del proyecto se tomó en cuenta el tiempo y recursos que se utilizó en el mismo, servidores no fueron contados como parte de costos ya que los que se usan son de propiedad del sistema RAP y solo son reutilizados en la creación de este módulo.

#### 3.2.1 Análisis de requerimientos

Antes de empezar el prototipado o el desarrollo del proyecto, en esta sección se destinaron 12 horas. Estas horas de trabajo comprendieron la recopilación de datos, la evaluación de requisitos y la definición clara de los objetivos del proyecto. Siendo coherentes con la tarifa establecida, el costo asociado a esta fase es de \$138.00 (12 horas \* \$11.50/hora).

### **3.2.2 Prototipado**

El prototipado fue importante para mostrar al cliente como se llevaría el desarrollo del proyecto por lo que se usaron habilidades técnicas para la creación de una representación inicial de la solución propuesta. Se asignaron 8 horas a esta fase, durante las cuales se trabajó en la conceptualización y creación de un prototipo funcional. El costo total del prototipado se calcula en \$92.00 (8 horas \* \$11.50/hora).

### **3.2.3 Desarrollo de Código**

Luego de la aprobación de prototipo por parte de los clientes la etapa de desarrollo del código se empezó, donde se transforman conceptos en una solución, para los servidores no se invirtió un adicional ya que el módulo se entrega a los usuarios del CTI y usan servidores ya obtenidos del RAP. En esta sección se invirtieron un total de 70 horas en esta fase, dedicadas al diseño. Codificación, pruebas y optimización del software. Con la tarifa establecida de \$11.50 por hora los costos acumulados para el desarrollo del código ascienden a \$805.00 (70 horas \* \$11.50/hora).

### **3.2.4 Total de costos**

La suma de los gastos incurridos en el desarrollo de la solución, tomando en cuenta el análisis del proyecto, prototipado y desarrollo del código representa un costo de \$1,035.00. Adicionalmente, se deben incluir 20 horas adicionales para documentación, lo que eleva el costo total a \$1,265.00.

**Tabla 3.1***Costos de implementación*

<b>Componente</b>	<b>Horas de trabajo</b>	<b>Costo por hora</b>	<b>Costo total</b>
<b>Análisis de requerimientos</b>	12	\$11.50	\$138.00
<b>Prototipado</b>	8	\$11.50	\$92.00
<b>Desarrollo de Código</b>	70	\$11.50	\$805.00
<b>Documentación</b>	20	\$11.50	\$230.00
<b>Total</b>	110		\$1,265.00

*Nota. Estos son costos estimados de lo que costó implementar el proyecto.*

### **3.3 Entregables**

Módulo de etiquetado, una versión bifurcada del proyecto original que incluye el módulo de etiquetado manual, acceso al mismo desde el sistema RAP y los cambios implementados según lo acordado.

Documentación Detallada en el README, donde se proporciona documentación exhaustiva en el archivo README del repositorio. Esta documentación describe en detalle los cambios efectuados en los archivos del plugin, registros detallados de las modificaciones realizadas en los archivos originales del proyecto. Además de instrucciones de como levantar el proyecto de manera local y las dependencias necesarias.

## **4. Capítulo 4**

## 4.1 Conclusiones y recomendaciones

### 4.1.1 Conclusiones

- La implementación del módulo de etiquetado de gestos y posturas dentro del sistema RAP se logró con éxito con el uso del framework *Django*, módulo que resulta fundamental para mejorar la precisión y calidad del etiquetamiento de videos del modelo del RAP.
- Una fortaleza del proyecto fue la interfaz intuitiva que facilita a los usuarios acceder al módulo desde el sistema rap sin tener que migrar videos o cambiar de plataforma, ya que se muestra una lista de videos que necesitan etiquetado manual.
- Los resultados demuestran la viabilidad y eficacia del módulo de etiquetado manual, siendo una implementación con funciones exitosas, que son, la creación y eliminación de etiquetas en tiempo real, junto con la generación de las etiquetas guardadas en un archivo CSV, respalda su utilidad en el proceso de etiquetado. Concluyendo que los objetivos planteados al inicio del proyecto, la implementación exitosa del módulo de etiquetamiento manual, así como la creación y eliminación de etiquetas, han cumplido con las metas establecidas.
- Al comparar con otras interfaces de etiquetamiento manual existentes, este proyecto destaca por integrarse en la página principal del sistema RAP, flexible y de fácil acceso. Además, la generación de archivos CSV resalta la relevancia en el ámbito de etiquetamiento de videos.

#### 4.1.2 Recomendaciones

- Se sugiere integrar el módulo de etiquetado con el servidor donde se realiza el reentrenamiento del modelo de inteligencia artificial. Esto permitirá mejorar el sistema RAP automatizando el proceso de envío del etiquetado al servidor de reentrenamiento.
- Se recomienda usar el repositorio del plugin *Video.js Annotation Comments* si se van a realizar grandes modificaciones del módulo de etiquetado. El repositorio permite cambiar el funcionamiento del plugin con mayor facilidad y generar los archivos JS correspondientes, en vez de modificar directamente los archivos JS ya existentes del plugin. Para poder modificar el plugin desde el repositorio original, se debe realizar una tarea de ingeniería inversa para poder identificar los componentes de los archivos JS y adaptarlos al repositorio. Cuando se tengan los cambios adaptados, se pueden empezar a implementar las nuevas modificaciones requeridas.
- Se sugiere añadir al CSV generado la información de los frames que tienen cada segundo, ya que los videos pueden tener distintas tasas de frames por segundo.

## Referencias

- [1] E. Gaur, V. Saxena and S. K. Singh, "Video annotation tools: A Review," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 911-914, doi: 10.1109/ICACCCN.2018.8748669.
- [2] C. Vondrick, D. Patterson, D. Ramanan, "Efficiently Scaling Up Crowdsourced Video Annotation", Proceedings of International Journal of Computer Vision, pp. 184–204, 2013.
- [3] S. Bianco, G. Ciocca, P., Napoletano, R.Schettini, "An interactive tool for manual, semi-automatic and automatic video annotation", Proceedings of Computer Vision and Image Understanding, vol. 131, pp. 88-99, 2015.
- [4] D. Sgouropoulos. "Annotator." GitHub.com.  
<https://github.com/dsgou/annotator> (accedido jun. 6, 2023).
- [5] S. Firat. "bilkent video annotation tool". GitHub.com.  
<https://github.com/selimfirat/bilkent-video-annotation-tool> (accedido jun. 6, 2023).
- [6] Contently. "videojs annotation comments". GitHub.com.  
<https://github.com/contently/videojs-annotation-comments> (accedido jun. 21, 2023).
- [7] Video.js. "video.js". GitHub.com.  
<https://github.com/videojs/video.js> (accedido jun. 21, 2023).
- [8] Django Software Foundation. "How to create CSV output". djangoproject.com.  
<https://docs.djangoproject.com/en/4.2/howto/outputting-csv/> (accedido jun. 21, 2023).
- [9] R. M. Zevallos. "Etiquetado RAP". <https://marvelapp.com>.  
<https://marvelapp.com/prototype/2jcc6ic6> (accedido jul. 5, 2023).

## A. APÉNDICE

## Requerimientos funcionales

### Figura A.1

#### *Requerimientos del proyecto*

Requerimientos del proyecto

El proyecto de etiquetado de posturas contará con los siguientes requerimientos:

- 1.El módulo de etiquetamiento manual de posturas debe ser implementado utilizando el framework Django.
- 2.El módulo debe ser compatible e integrable con el sistema RAP.
- 3.El sistema RAP debe tener una funcionalidad para identificar videos con errores de etiquetamiento automático.
- 4.El sistema RAP debe ser capaz de generar y mostrar una lista de videos que presenten esos errores.
- 5.El sistema RAP debe permitir únicamente a los usuarios designados acceder al módulo de etiquetamiento de posturas desde los videos que presentan errores.
- 6.El módulo debe proporcionar opciones para etiquetar gestos y posturas del video seleccionado mediante alguna herramienta de anotación de video.
- 7.La herramienta de anotación de video debe permitir seleccionar una sección de tiempo del video correspondiente y asignarle un gesto o postura.
- 8.El sistema debe generar archivos CSV que contengan la información del etiquetado manual.
- 9.Los archivos CSV deben incluir las líneas de tiempo correspondientes a los gestos y posturas etiquetados del video seleccionado.

  
LEONARDO DAVID ERAS  
DELEGADO

---

Firma del cliente

Nota. Archivo PDF firmado por el cliente.

## Flujo de ventanas

### Figura A.2

Documento con flujo de ventanas parte 1

#### Flujo de ventanas del proyecto

**Ver tema**

**Editar tema**

Uso del lenguaje Rust en programación de sistemas embebidos, ¿reemplazará a C/C++?

- Ver Horarios
- Desactivar tema
- Asociar Encuesta
- Guardar

**Listado de resultados**

Cédula	Nombre	Apellidos	Ver	Revisar	Procesar
XXXXXXXXXX	OSCAR EDUARDO	VITE POMBAR	Ver	Revisar	Procesar
XXXXXXXXXX	NAN FERNANDO	BUENDIA GARCIA	Ver	Revisar	Procesar
XXXXXXXXXX	ANTHONY LEONARDO	DELGADO ESPINEL	Ver	Revisar	Procesar
XXXXXXXXXX	JOSEPH MAVERICK	PIGAÑO APOLINARIO	Ver	Revisar	Procesar
XXXXXXXXXX	LUIS ALBERTO	ANDRADE PROAÑO	Ver	Revisar	Procesar
XXXXXXXXXX	CARLOS ISRAEL	JIMENEZ JIMENEZ	Ver	Revisar	Procesar
XXXXXXXXXX	JOSE ANTONIO	MORAN RODRIGUEZ	Ver	Revisar	Procesar
XXXXXXXXXX	LEON FELIPE	VARGAS AGUILAR	Ver	Revisar	Procesar
XXXXXXXXXX	DARIANE ELISA	MEDRANDA MACHADO	Ver	Revisar	Procesar
XXXXXXXXXX	ALEJANDRA DIONE	ALAVA MUÑOZ	Ver	Revisar	Procesar
XXXXXXXXXX	JOEL RONALDO	TAMAYO BRAVO	Ver	Revisar	Procesar
XXXXXXXXXX	PEDRO ALEJANDRO	RENDON PIMTOJA	Ver	Revisar	Procesar

**Promedio de resultados de usuarios que han realizado el tema "Uso del lenguaje Rust en programación de sistemas embebidos, ¿reemplazará a C/C++?"**

Calificación global

Puntajes

Miradas

Pausas sonoras

Visitas

Dispositivos

Promedio de características de resultados

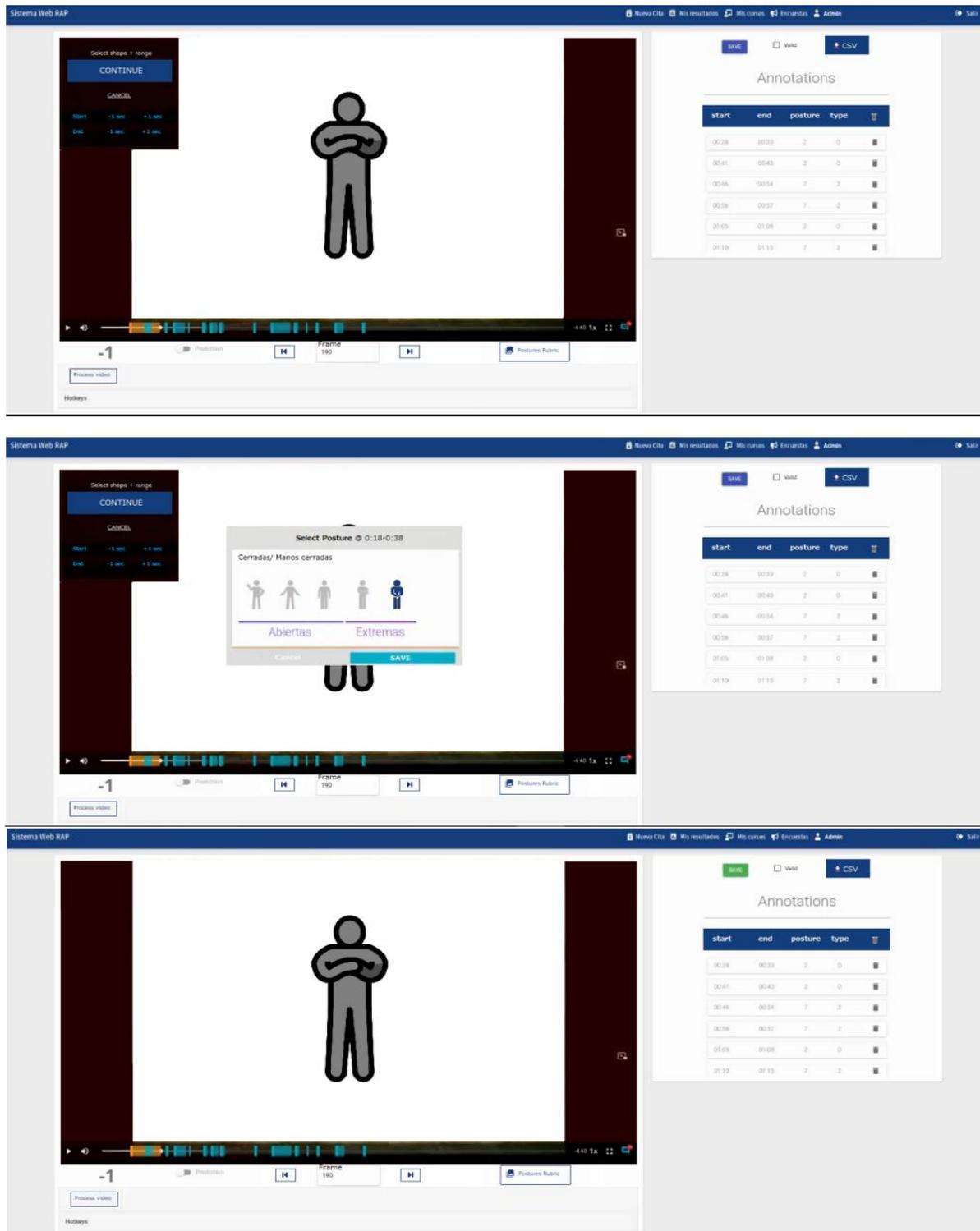
**Annotations**

start	end	posture	type	
00:28	00:33	2	0	
00:41	00:43	2	0	
00:46	00:54	7	2	
00:56	00:57	7	2	
01:05	01:06	2	0	
01:10	01:13	7	2	

Nota. Primera página del documento PDF firmado.

Figura A.3

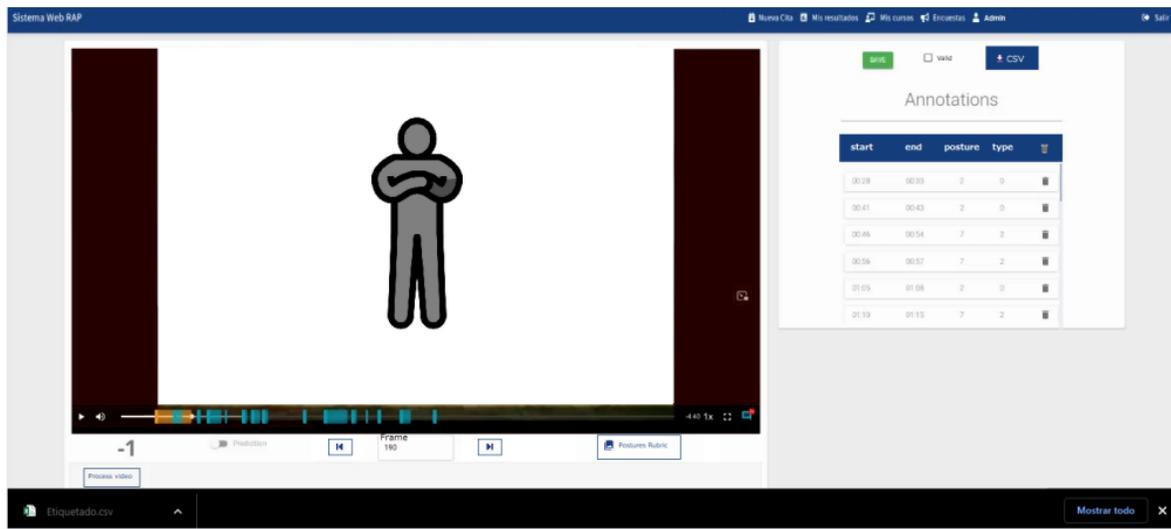
Documento con flujo de ventanas parte 2



Nota. Segunda página del documento PDF firmado.

## Figura A.4

Documento con flujo de ventanas parte 3



Link del flujo de ventanas en marvelapp: <https://marvelapp.com/prototype/2jcc6ic6>



Documento firmado por  
LEONARDO DAVID ERAS  
DELGADO

Firma del cliente

Nota. Última página del documento PDF firmado.

## Documentación del proyecto

### Figura A.5

#### Documentación del proyecto parte 1

Readme.md

## Proyecto RAP - Etiquetado con VideoJS Annotation Comments

Este README proporciona una guía detallada para configurar el sistema de etiquetado utilizando el plugin VideoJS Annotation Comments en el proyecto RAP.

### Instalación y Configuración

#### 1. Instalación del Sistema RAP Base

Una vez instalado el sistema RAP base, procede a aplicar las nuevas migraciones:

```
python manage.py makemigrations
python manage.py migrate
```

#### 2. Configuración del Etiquetado

Para que el etiquetado funcione correctamente, instala las dependencias de npm en la carpeta "static":

```
npm install video.js@7
npm install @contently/videojs-annotation-comments
```

Luego, reemplaza el contenido de la carpeta "videojs-annotation-comments" en el repositorio con los archivos correspondientes desde "node\_modules":

```
cp -r videojs-annotation-comments/* static/node_modules/@contently/videojs-annotation-comments
```

### Cambios Realizados en el Código

#### Archivos Relevantes de Etiquetado

- **models.py**: Actualizaciones en la base de datos relacionadas con el etiquetado.
- **urls.py**: Modificación de 5 URLs para las funciones del backend.
- **views.py**: Cambios en funciones del backend, incluyendo:
  - `etiquetas_csv()`: Genera un archivo CSV con las etiquetas.
  - `format_two_digits()`: Función auxiliar para `get_annotation_list()`.
  - `get_annotation_list()`: Obtiene etiquetas de un resultado concreto.
  - `etiquetado_manual()`: Crea etiquetas para inicializar el plugin.
  - `eliminar_etiqueta()`: Elimina una etiqueta de la base de datos.
  - `nueva_etiqueta()`: Añade una nueva etiqueta en la base de datos.

#### Cambios en Archivos Estáticos (Static)

- Se añadió la carpeta "fonts" y un archivo de CSS relacionados con "glyphther".
- Ejecuta el comando "npm install" en "static" para instalar las librerías necesarias.

## Figura A.6

### Documentación del proyecto parte 2

#### Modificaciones en "videojs annotation comments"

Se realizaron varias modificaciones en los archivos del plugin "videojs annotation comments" (ubicados en `static/node_modules/@contently/videojs-annotation-comments`) para lograr la implementación deseada del módulo de etiquetado. Estas modificaciones se realizaron en los archivos de videojs annotation comments ya modificados del módulo de etiquetado en el que nos estamos basando ([enlace](#)) para esta implementación. A continuación, se detallan las modificaciones realizadas en las líneas de código correspondientes:

1. Modificación en `handlebars` (archivo: `videojs-annotation-comments.js`, línea 6325):
  - **Cambios:** Modificado el primer return de `handlebars` para eliminar la fecha.
  - **Descripción:** Esto elimina la fecha que se muestra junto al comentario en la parte derecha del reproductor de video.
2. Eliminación de botón "ADD REPLY" (archivo: `videojs-annotation-comments.js`, línea 6342):
  - **Cambios:** Se borró el botón "ADD REPLY".
  - **Descripción:** El botón "ADD REPLY" se eliminó ya que su implementación no era adecuada.
3. Modificación en retorno HTML de la función "9" (archivo: `videojs-annotation-comments.js`, línea 6398):
  - **Cambios:** Modificaciones en el retorno HTML de la función "9".
  - **Descripción:** Se realizaron ajustes en el HTML generado por la función "9" de Handlebars, que es responsable de proporcionar el contenido del cuadro de creación de etiquetas y selección de posturas/miradas.
4. Agregada función `addNewAnnotation` (archivo: `videojs-annotation-comments.js`, línea 6771):
  - **Cambios:** Añadida la función `addNewAnnotation` para registrar la creación de anotaciones.
  - **Descripción:** Esta función se utiliza para que el plugin pueda comunicarse con Django y notificar cuando se crea una anotación, permitiendo obtener información relevante sobre dicha anotación.
5. Función `selectMirada` (archivo: `videojs-annotation-comments.js`, línea 7722):
  - **Cambios:** Implementación de la función `selectMirada` para reconocer la selección de miradas.
  - **Descripción:** Esta función se encarga de identificar qué mirada ha sido seleccionada en el módulo de etiquetado.
6. Modificación en función `bindEvents` (archivo: `videojs-annotation-comments.js`, línea 7515):
  - **Cambios:** Ajustes en la función `bindEvents` para limitar intervalos de tiempo.
  - **Descripción:** Estos cambios garantizan que los botones siempre respeten un rango de 1 segundo y no se extiendan más allá de los límites del video.

## Figura A.7

### Documentación del proyecto parte 3

7. Agregado evento en función `selectMirada` (archivo: `videojs-annotation-comments.js`, línea 7557):

- **Cambios:** Se agregó un listener/evento en la función `selectMirada`.
- **Descripción:** Este evento se encarga de gestionar la selección de miradas y realizar acciones correspondientes.

8. Modificación en función `startAddNew` (archivo: `videojs-annotation-comments.js`, línea 7771):

- **Cambios:** Modificaciones en la función `startAddNew` para asegurar un intervalo mínimo de 1 segundo.
- **Descripción:** Esto garantiza que el intervalo seleccionable para la edición siempre sea al menos de 1 segundo y evita situaciones no deseadas.

9. Modificación en función `draggableMarker` (archivo: `videojs-annotation-comments.js`, línea 8058):

- **Cambios:** Ajustes en la función `ondrag` dentro de `draggableMarker` para limitar selecciones.
- **Descripción:** Estos cambios evitan que se pueda seleccionar un segundo concreto, garantizando un comportamiento controlado.

Estas modificaciones se realizaron con el propósito de implementar el módulo de etiquetado según los requerimientos específicos del proyecto RAP.

#### Creación de la Template de Etiquetado

- Creación de una template que contiene el módulo de etiquetado.
- Archivos "etiquetado.css" en "static/css" y "etiquetado.js" en "static/js".
- En "etiquetado.js", funciones para inicializar video.js, enviar anotaciones al backend, registrar eventos y obtener listas de anotaciones.

#### Actualizaciones en Templates

- Agregados enlaces a archivos CSS y JS en "base.html".
- Modificación en "lista.html" para permitir acceso al módulo de etiquetado desde resultados.

#### ¡Listo para Etiquetar!

Ahora has completado la configuración y los cambios necesarios para habilitar la funcionalidad de etiquetado el sistema RAP.