Facultad de Ingeniería
Universidad de la República

Generador automático de índices en videos

Informe de Proyecto de Grado presentado al Tribunal Evaluador como requisito de graduación de la carrera Ingeniería en Computación

Documentación del trabajo realizada por:

María Jimena Campiotti
Leonardo Oyharzabal
Bruno Schiaffino
María Belén Taboas

Tutor: Antonio López
Cliente: Agustín Pereira

Montevideo, Uruguay · setiembre 2017 - setiembre 2018

Resumen

Cuando una persona quiere localizar cierto contenido en un libro, recurre a su índice sin dudarlo; sería algo poco común que alguien recorriera página por página hasta encontrar lo que busca. Sin embargo, al pensar en un video, resulta normal tener que recorrer su línea de tiempo para ubicar el momento que se quiere ver. Por otro lado, teniendo en cuenta el enorme crecimiento de las plataformas de videos actuales, se torna inviable identificar los momentos de interés de forma manual para la gran cantidad de videos disponibles.

Hoy en día, el amplio desarrollo en el campo de la inteligencia artificial permite que una máquina imite las funciones cognitivas de los humanos, como aprender o resolver problemas. En particular, existen herramientas que logran reconocer el contenido de videos, imágenes y audio. Esta tarea de reconocimiento es precisamente la que una persona debería ejecutar manualmente para la creación de un índice.

Este proyecto persigue justamente ese objetivo, emular lo que un humano haría a la hora de construir un índice para un video, con la ventaja de realizarlo de forma automática. En consecuencia, se desarrolla un prototipo de software que analiza videos en busca de objetos, personas, acciones, lugares, diálogos y toda clase de información que pueda ser de interés para un usuario. Todo esto, consumiendo servicios provistos por gigantes de la industria tales como Google o Amazon. Con la información obtenida se produce un conjunto de etiquetas asociadas a instantes de tiempo en el video, conformando así, un índice.

Adicionalmente, se implementan mecanismos que pretenden elevar la calidad de las etiquetas generadas. Esto se hace, en primer lugar, a través del filtrado de aquellas que son intrascendentes o erróneas. Sumado a esto, se utiliza el servicio de búsquedas de Google para determinar la relevancia de una etiqueta en base a la cantidad de resultados obtenidos para su búsqueda, descartando las que retornen pocos resultados. Por último, se busca fortalecer el prototipo utilizando la retroalimentación de los usuarios. Estos pueden indicarle al sistema las etiquetas que le resulten más interesantes para cierta categoría de video y así, se dejan sentadas las bases para explotar esta información en trabajos a futuro.

Finalmente, los resultados obtenidos reflejan que el objetivo de crear índices para videos de forma automática se cumplió satisfactoriamente. La combinación y el procesamiento adecuado de la información adquirida permite identificar etiquetas que resultan útiles a la hora de explorar el contenido de un video. Luego, las aplicaciones que consuman estos resultados podrán presentar los índices para que los usuarios se sirvan de ellos. Asimismo, se detecta un gran potencial para la expansión del prototipo con el fin de enriquecer las etiquetas que forman parte de los índices.

Palabras clave: video, índice, automático, hito, etiqueta, generación, procesamiento, detección, inteligencia artificial.

Agradecimientos

El presente proyecto ha requerido de un gran esfuerzo tanto del equipo que lo llevó adelante como de quienes han colaborado en distintos aspectos del mismo.

Es por este motivo que se quiere agradecer, en primer lugar, a los tutores, quienes han sido de inmensa ayuda aportando su conocimiento, brindando consejos y sirviendo de guías para llevar a cabo el presente proyecto.

A las empresas Google y Clarifai, que otorgaron financiamiento para poder utilizar sus herramientas y de esta manera, posibilitaron el desarrollo de una solución de mayor calidad.

Y por último, a familiares, amigos y a todos aquellos que de una forma u otra han colaborado o estado presentes durante la elaboración de este trabajo.

Índice general

Índice de cuadros

Índice de figuras

Glosario

Algoritmo MD5 (Message-Digest Algorithm 5) Es un algoritmo de cifrado de 128 bits. Es comúnmente usado para verificar que un archivo haya sido alterado.

Amazon S3 Servicio de almacenamiento de archivos en la nube ofrecido por Amazon.

API (Application Programming Interface) Conjunto de reglas expuestas por un sistema de software para acceder a sus funcionalidades y/o procedimientos desde sistemas externos.

Back-end Término que hace referencia a la parte del sistema de software que contiene la lógica de negocio del mismo.

Biblioteca En informática, una biblioteca (o librería), es un conjunto de subprogramas utilizados para desarrollar software.

CSS (Cascading Style Sheets) Lenguaje de diseño gráfico utilizado para la definición y creación de estilos de un documento HTML.

Flac (Free Lossless Audio Codec) Forma de codificación de audio que permite la reducción del mismo sin pérdida de información. Los archivos de audio que utilizan esta codificación tienen extensión .flac.

Framework Conjunto estandarizado de conceptos, prácticas y criterios que sirven como referencia para abordar una problemática particular.

Front-end Capa de presentación de un sistema de software mediante la cual los usuarios interactúan con el mismo.

Google Cloud Platform Conjunto de servicios en la nube ofrecidos por Google. Provee diversos módulos entre los que se incluyen capacidad de cómputo, almacenamiento, análisis de datos y herramientas de aprendizaje automático.

Google Cloud Storage Servicio de almacenamiento de archivos en la nube de Google.

HTML (HyperText Markup Language) Lenguaje de computación utilizado para la creación de páginas web.

HTTP (HyperText Transfer Protocol) Protocolo de comunicación utilizado por la World Wide Web (WWW) que define cuál es el formato de los mensajes, cómo estos deben transmitirse y cuáles son las acciones que deben tomar los servidores web y navegadores frente a ciertos comandos.

Inteligencia Artificial Simulación de procesos de inteligencia humana por parte de máquinas, específicamente sistemas informáticos.

JSON (JavaScript Object Notation) Formato de texto ligero utilizado por sistemas de información para el intercambio de datos.

KPI (Key Performance Indicators) Medida del nivel del rendimiento de un proceso.

Microsoft Cognitive Services Conjunto de algoritmos de aprendizaje automático desarrollados por Microsoft para resolver problemas en el campo de la inteligencia artificial.

Path Ruta de un archivo o directorio en el sistema de archivos de un ordenador.

REST (Representational State Transfer) Estilo de arquitectura utilizado para el diseño de servicios web.

XML (Extensible Markup Language) Es un metalenguaje (lenguaje que se utiliza para definir otro) extensible basado en etiquetas.

Capítulo 1

Introducción

En el presente documento se expone el proceso y resultados del trabajo realizado en el marco de la asignatura Proyecto de Grado de Ingeniería en Computación. Se detallan las motivaciones que dieron pie a este proyecto, los desafíos afrontados durante el desarrollo, y las conclusiones y reflexiones a las que se llegaron al final del camino.

1.1. Motivaciones

Día a día, la cantidad de videos en internet crece de manera exponencial. Este crecimiento se extiende a través de múltiples sectores, desde el entretenimiento, deportes y publicidad, hasta empresas y educación. Por ejemplo, en YouTube, el sitio web más grande dedicado a compartir videos, se suben 300 horas de contenido por minuto. El número total de horas de videos vistas por día en YouTube es más de 5 billones. [1]

La amplia variedad de videos distribuidos en numerosas plataformas online hace que la búsqueda de contenidos se pueda tornar compleja. A su vez, en la mayoría de los casos, los videos disponibles en internet constan únicamente de un título y una descripción para ser identificados. Sin embargo, su contenido suele abarcar más de lo que logran capturar estos datos. Esta falta de información hace que la búsqueda de los momentos más relevantes dentro de un video sea una tarea tediosa.

A diferencia de los libros, en los videos no existe el concepto de índice. Por lo que para encontrar cierto instante de interés no queda otra opción más que recorrer la línea de tiempo hasta hallar lo que se está buscando. Para descubrir el contenido de un video, una persona debe reproducirlo y registrar manualmente los momentos que considera importantes. Generar dicha información es costoso, y para la extensa cantidad de videos existentes en internet se torna humanamente imposible.

En ese contexto, surgen dos proyectos diferentes dentro del marco de la asignatura Proyecto de Grado. El primero, comenzado a principios del año 2017, brinda a los usuarios una plataforma que les permite observar un video de YouTube y almcenar los hitos que le resulten de interés. De esta forma, en futuras visualizaciones, se evita recorrer la línea de tiempo en busca de los momentos deseados ya que el índice fue creado previamente para ese video. Este proyecto de

ahora en más será llamado Video++1.

El otro proyecto, comenzado a mitad del año 2017, es el que será presenteado en este informe. Tiene como objetivo eliminar el paso manual para generar un índice sobre un video, buscando que esto se haga de forma automática. Además, servirá de fuente de información para el proyecto Video++1, formando en conjunto una plataforma en donde un usuario puede obtener índices de manera automática para cualquier video. Este proyecto tiene el nombre de "Generador automático de índices en videos" y será referenciado durante el presente informe como Video++2.

1.2. Descripción del proyecto

Este trabajo pretende abordar la problemática de la búsqueda de hitos dentro de un video. Se entiende por hito a la dupla <etiqueta, instante de tiempo>, donde el segundo campo se refiere al momento en el que dicha etiqueta aparece en el video. A lo largo del presente informe se utilizarán de manera indistinta las palabras índice e índices en referencia al conjunto de hitos reconocidos.

Para la resolución del problema, se propone la creación de un prototipo de software capaz de recolectar la información necesaria para la creación del índice de un video. Con esa premisa, se estudian diversas herramientas de procesamiento de imágenes, video, y audio. Estas se utilizan luego, de manera conjunta, para lograr identificar elementos dentro del video en diversos instantes de tiempo y así poder generar un índice sin la necesidad de intervención humana.

1.3. Objetivos

1.3.1. Objetivos generales

- Realizar un estudio de las herramientas de inteligencia artificial para ser utilizadas en el sistema a construir.
- Crear un prototipo de software que permita identificar entidades y acciones, dentro de un video online, junto con los instantes de tiempo en los que aparecen.
- Mediante la información obtenida en el punto anterior, construir índices de manera automática sobre dichos videos.

1.3.2. Objetivos específicos

- Servir de fuente de información para el proyecto Video++1, permitiendo iniciar la creación de índices desde dicha aplicación.
- Generar mecanismos que mejoren la calidad de los índices creados, para aportar mayor valor a los usuarios que los consumen.

- Construir una fuente de información que pueda ser explotada en el futuro, con el fin de mejorar los índices creados a través de algoritmos más sofisticados.

1.4. Estructura del documento

En el primer y actual capítulo, se describe el proyecto, las motivaciones que dieron lugar a su desarrollo, y principales objetivos.

En el segundo capítulo, se presenta el resultado del relevamiento de información sobre el estado tecnológico actual del dominio del problema.

En el tercer capítulo, se muestran los resultados de las pruebas de concepto que fueron realizadas sobre las diferentes herramientas investigadas, con el fin de utilizarlas en la solución desarrollada.

En el cuarto capítulo, se expone la especificación funcional del sistema. Se detalla tanto el alcance como los requerimientos funcionales y no funcionales.

En el quinto capítulo, se describe el diseño de la solución. Se detalla la arquitectura, principales componentes, y la interacción entre ellos.

En el sexto capítulo, se define el modelo de datos a utilizar para representar la información manejada por el sistema.

En el séptimo capítulo, se detalla la implementación del sistema y cómo se resolvieron, a bajo nivel, sus diferentes funcionalidades.

En el octavo capítulo, se deja constancia de las pruebas realizadas sobre el software desarrollado y las conclusiones obtenidas a partir de las mismas.

En el noveno capítulo, se describe el proceso de gestión del proyecto. Se exploran aspectos relacionados a planificación y modalidad de trabajo.

Finalmente, en el décimo capítulo, se presentan las conclusiones del proyecto, así como también el posible trabajo a futuro.

Capítulo 2

Estado del arte

En el presente capítulo se introduce el estado actual del conocimiento sobre la problemática abordada. Se hace un recorrido por las diferentes tecnologías existentes que, de alguna forma u otra, contribuyen a la solución de indexado automático de videos.

En primer lugar, se introduce al lector sobre algunas nociones básicas de algoritmos de aprendizaje automático. Esta disciplina es la base de las herramientas que se presentarán luego.

En segundo lugar, se estudian diversos proveedores de tecnologías que a través de dichos algoritmos, permiten analizar tanto videos, como imágenes y audio.

Finalmente, se concluye acerca de las herramientas investigadas y cómo estas pueden contribuir a la solución de la problemática planteada en este proyecto.

2.1. Aprendizaje automático

Aprendizaje automático, en inglés *machine learning*, es una rama dentro del campo de la inteligencia artificial (IA). Su objetivo consiste en el desarrollo de técnicas que permitan a las computadoras aprender sin ser programadas de forma explícita. Para ello, se utilizan algoritmos capaces de generalizar comportamientos a partir de ejemplos suministrados. El aprendizaje automático toma dichos datos para detectar patrones, y ajustar las acciones del programa en consecuencia. Es por lo tanto, un proceso de inducción de conocimiento.

Existen diversos algoritmos de aprendizaje automático, a continuación se pasa a describir los más tradicionales.

- **Aprendizaje supervisado:** Su objetivo es generar una función capaz de predecir el valor de salida correspondiente para cualquier elemento de entrada. Esto se logra a través del suministro de datos de entrenamiento, a partir de los cuales el programa intentará ajustar su función y brindar resultados que mejoren con el tiempo. De esta manera, luego de someter al programa a un volumen considerable de datos, este será capaz de aprender de los ejemplos suministrados y predecir salidas para situaciones no vistas previamente.
- **Aprendizaje no supervisado:** A diferencia del aprendizaje supervisado, este método consiste en suministrar al sistema únicamente datos de entrada. Estos no se encuentran

clasificados ni tampoco se sabe de antemano cuál es la salida esperada. De esta manera, el programa deberá ser capaz de estudiar las propiedades de los elementos de entrada, para luego reconocer patrones y brindar los resultados deseados.

- **Aprendizaje por refuerzo:** A grandes rasgos, este algoritmo aprende observando el mundo que le rodea. Su información de entrada, consta de la retroalimentación que obtiene del entorno en respuesta a sus acciones. Por lo tanto, se puede decir que el sistema utiliza una técnica a base de ensayo y error. De esta manera, cuando el sistema produce un resultado correcto, un agente externo debe recompensarlo (reforzarlo) para que este comportamiento se repita a futuro. Por el contrario, si se produce un resultado incorrecto, el programa será "castigado" y buscará no repetir el comportamiento. Es así como el sistema va ajustando su función a modo de mejorar las respuestas brindadas.

2.1.1. Aplicaciones

La cantidad de aplicaciones para el aprendizaje automático crece de forma exponencial cada día. Un escenario cotidiano de su uso se da en las redes sociales que tanta fuerza han cobrado en el presente. Por ejemplo, Facebook lo utiliza para personalizar la página de inicio de cada miembro. Si un individuo se detiene para leer o dar "me gusta" a las publicaciones de cierto amigo, su página principal comenzará a mostrar más actividad de ese amigo en particular. Por detrás, el software utiliza algoritmos de aprendizaje automático para identificar patrones en el comportamiento de los usuarios y así, adaptar las noticias desplegadas a los gustos e intereses de cada uno.

Sin ir muy lejos, existen diversas empresas en Uruguay que trabajan en esta rama de la IA. Tal es el caso de Tryolabs, una consultora de *machine learning* que tiene como objetivo ayudar a otras empresas a crear soluciones basadas en la recopilación de datos para poder mejorar su KPI.

2.1.2. ¿Por qué aprendizaje automático para la generación de índices?

Como se mencionó en la introducción, la construcción manual de índices sobre videos se puede tornar una tarea compleja y tediosa debido al tiempo y esfuerzo que llevaría completarla. Es aquí, donde el aprendizaje automático entra en juego.

El procesamiento de videos para el reconocimiento de objetos o acciones sin la necesidad de intervención humana, requiere del uso de sistemas que hayan sido entrenados para poder determinar lo que se está observando. Utilizando técnicas de aprendizaje automático, se le pueden enseñar al sistema las propiedades de los elementos para que este pueda identificarlos. Entonces, a medida que el programa se va ejercitando, logrará reconocer en el video los objetos o acciones que aparecen en cierto instante del tiempo.

En el contexto de este proyecto, no se desarrollarán algoritmos de aprendizaje automático,

ya que esto escapa de los objetivos perseguidos. El motivo, es que el dominio de la técnica para implementarlos requiere de conocimientos que llevan meses de estudio y esto, excede al alcance planteado.

En esta oportunidad, lo que se busca es hacer uso de servicios que, utilizando algoritmos de aprendizaje automático, puedan extraer información sobre los videos procesados. Estos datos se utilizarán finalmente para generar los índices, siendo el sistema el encargado de reconocer los objetos, y no un humano.

2.2. Herramientas de procesamiento de multimedia

2.2.1. Google

Google pone a disposición un conjunto de servicios web que encapsulan algoritmos de aprendizaje automático para facilitar el análisis de contenido en imágenes, videos y audios. Se dará una descripción de cada uno de ellos.

2.2.1.1. Vision API

Vision API permite identificar contenido en imágenes, pudiendo catalogarlas en miles de categorías. Es posible detectar amplios conjuntos de objetos, así como personas y animales. A su vez, dispone de una funcionalidad de detección de texto.

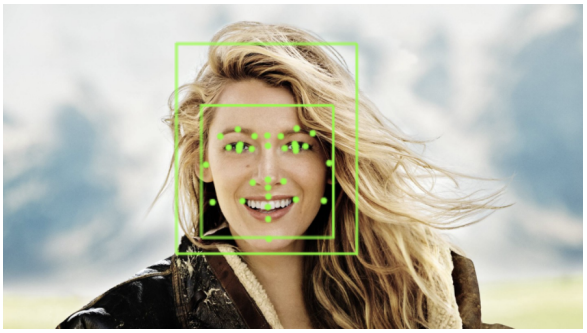
Mediante peticiones a esta API, se puede analizar una imagen concreta, extrayendo distintos tipos de información que se detallan a continuación.

Detección de rostros

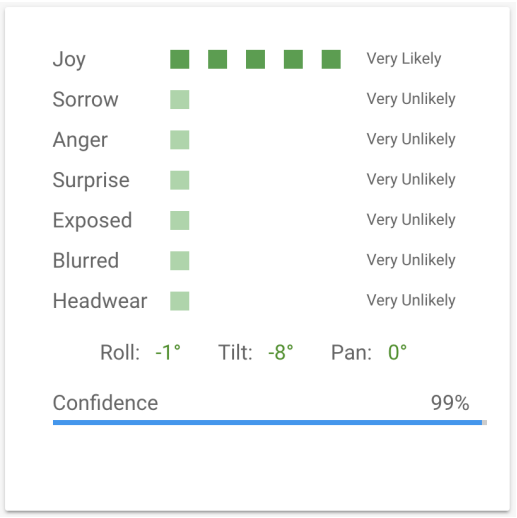
El objetivo de esta funcionalidad es la detección de rostros humanos en imágenes. También incluye información acerca de las coordenadas de la posición de la cara y puntos de referencia como ojos, nariz y boca, entre otros. Asimismo, permite identificar emociones tales como sorpresa, alegría, tristeza o enojo. La herramienta asigna un porcentaje de confianza a cada detección que realiza. Esto es una medida de qué tan probable es que se haya reconocido cierto elemento de forma acertada.

En la figura 2.1 se observan los resultados de esta funcionalidad para una imagen seleccionada como ejemplo.

(a) Imagen seleccionada para el análisis [24]



(b) Resultado gráfico del análisis

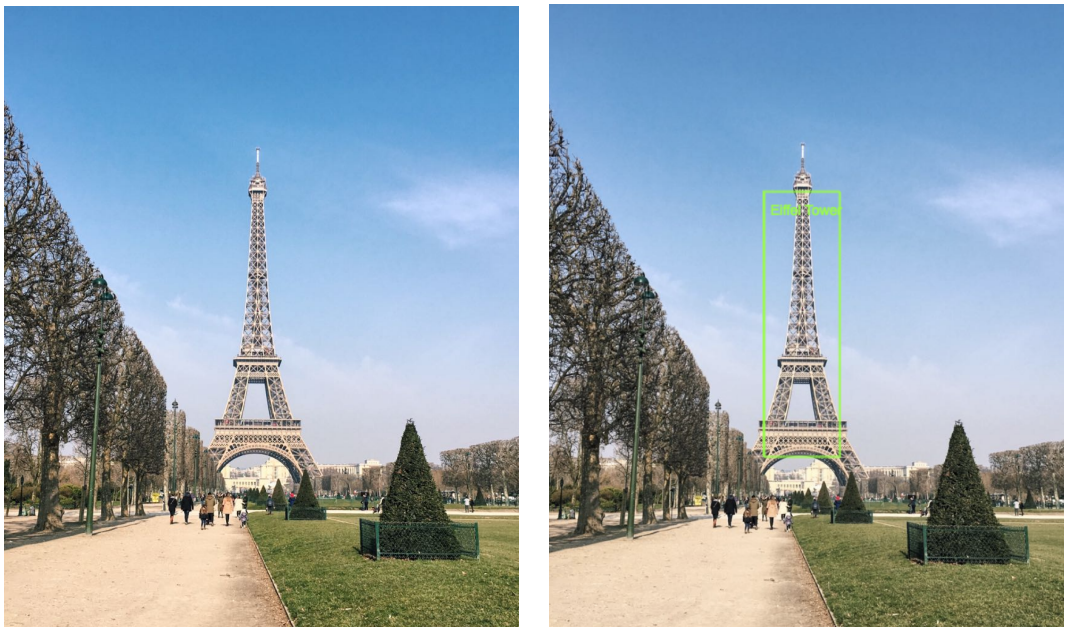


(c) Resultado final del análisis para detección de rostros

Figura 2.1: Análisis de detección de rostros de *Vision API*

Detección de puntos de referencia

Esta funcionalidad pretende identificar estructuras populares, tanto naturales como construidas por el hombre. A su vez, brinda información sobre la longitud y la latitud del punto de referencia identificado.



(a) Imagen seleccionada para el análisis

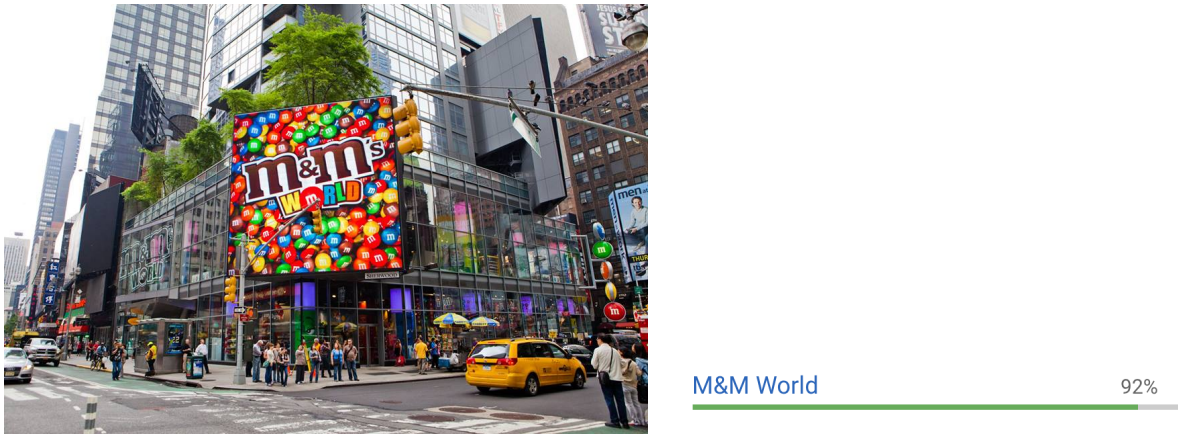
(b) Resultado del análisis

Figura 2.2: Detección de la Torre Eiffel utilizando la detección de puntos de referencia de *Vision API*

Detección de logos

Esta herramienta tiene como objetivo la identificación de logotipos conocidos. Además, retorna las coordenadas en donde se encuentra el logo dentro de la imagen analizada.

La figura 2.3b, muestra que la funcionalidad logra identificar el logo contenido en la imagen con una confianza del 92 %.



(a) Imagen seleccionada para el análisis de detección de logos [26]

(b) Resultado del análisis

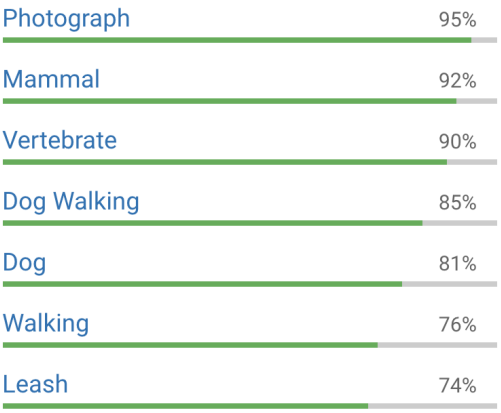
Figura 2.3: Detección del logo de la tienda M&M World, Nueva York

Detección de etiquetas

Detecta entidades de diversos conjuntos de categorías dentro de una imagen, entre las que se encuentran medios de transporte, animales y acciones.

La figura 2.4b, muestra las principales etiquetas reconocidas por la herramienta para la imagen seleccionada. Cada etiqueta, se despliega en inglés y tiene asociada un porcentaje de confianza. Por ejemplo, "dog walking" (paseando un perro) se encontró con una confianza del 85 %.

(a) Imagen seleccionada para el análisis [34]



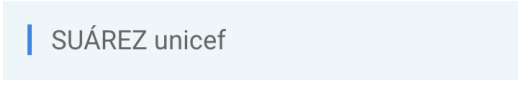
(b) Resultado del análisis para detección de etiquetas

Figura 2.4: Detección de etiquetas de *Vision API*

Reconocimiento de texto

Esta funcionalidad permite extraer texto embebido en una imagen. Es compatible con un gran número de idiomas. A continuación se presenta un ejemplo:

(a) Imagen seleccionada para el análisis [30]



(b) Texto reconocido

Figura 2.5: Detección de texto de *Vision API*

Detección de contenido explícito

Esta funcionalidad tiene como objetivo reconocer contenido para adultos, incluyendo aspectos como desnudez, actividad sexual o pornografía. La herramienta maneja cinco categorías de contenido explícito y evalúa qué tan probable es que la imagen se corresponda con alguna de ellas. Los valores de probabilidad se definen en cinco niveles: "muy improbable", "improbable", "posible", "probable" y "muy probable".

Las categorías de contenido explícito reconocidas por la herramienta se detallan a continuación:

- *Contenido adulto*: permite diferenciar entre las imágenes pornográficas y no pornográficas.
- *Contenido atrevido*: incluye posturas obscenas o provocadoras, ropa transparente o translúcida, primeros planos de regiones sensibles, entre otros.
- *Contenido violento*: relacionado con imágenes que representan asesinatos, armas, sangre, guerra, entre otros.
- *Contenido médico*: imágenes que contienen cirugías, enfermedades o partes del cuerpo.
- *Contenido paródico*: indica contenido que haya sido modificado del original a modo de burla u ofensa.

A continuación se presenta un ejemplo del resultado obtenido por esta funcionalidad:

(a) Imagen de ejemplo seleccionada [28] (b) Resultado del análisis

Figura 2.6: Detección de contenido explícito de *Vision API*

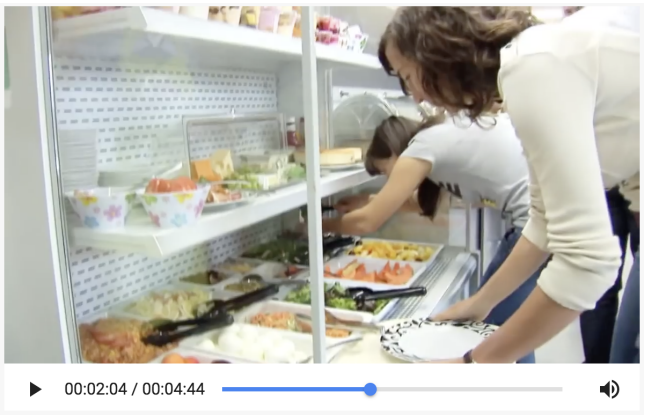
2.2.1.2. Cloud Video Intelligence

A través de diferentes APIs, *Cloud Video Intelligence* expone servicios para la extracción de contenido en videos. Entre las funcionalidades que ofrece se encuentran: identificación de entidades, contenido para adultos y detección de cambios de escenas. Estas se describen a continuación:[1]

Detección de etiquetas

De forma análoga a la funcionalidad de *Vision API*, permite etiquetar entidades detectadas en un video. Cada elemento identificado incluye el segmento de tiempo del video en donde este aparece.

La figura 2.7, presenta el resultado del análisis de detección de etiquetas para un video de ejemplo.



(a) Captura del video en el segundo 124


```
{
  "categoryEntities": [
    {
      "description": "food",
      "entityId": "/m/02wbm",
      "languageCode": "en-US"
    }
  ],
  "entity": {
    "description": "take out food",
    "entityId": "/m/01w53b",
    "languageCode": "en-US"
  },
  "segments": [
    {
      "confidence": 0.6800978,
      "segment": {
        "endTimeOffset": "131.164366s",
        "startTimeOffset": "124.424300s"
      }
    }
  ]
},
```


(b) Parte de la respuesta retornada por la API

Figura 2.7: Análisis de detección de etiquetas de *Cloud Video Intelligence*

En la imagen 2.7b se observa el formato en el que la API devuelve una etiqueta detectada. Por un lado, la clave `categoryEntities` indica la categoría a la cual pertenece la etiqueta encontrada, en este caso, "food" (comida). Por otro lado, `entity` contiene la información de la etiqueta propiamente dicha. En este caso `description` indica que se detectó "take out food" (comida para llevar).

Seguidamente, la sección `segments` comprende la confianza con la cual se detectó la etiqueta, así como también, los segmentos de tiempo (en segundos) en donde se encuentra en el video. Para corroborar esto, se observa en la figura 2.7a una captura del video en el instante de tiempo en el que aparece la etiqueta detectada.

[1]Todos los videos de ejemplo utilizados en esta sección se encuentran disponibles en `https://cloud.google.com/video-intelligence/`.

Detección de cambios de escenas

Esta funcionalidad analiza y detecta los cambios de escena que ocurren a lo largo de un video. Un cambio de escena se define como una transición de contenido. A modo de ejemplo, un video de un juego de golf siguiendo a dos jugadores a través del campo con algunas tomas panorámicas del bosque, produciría dos escenas que podrían ser las asociadas a los "jugadores" y al "bosque". Para cada cambio de escena identificado, la herramienta especifica su tiempo de inicio y fin.

Detección de contenido explícito

Al igual que la herramienta existente para imágenes, su objetivo es detectar contenido para adultos. Su funcionamiento es análogo a su contraparte, por lo que no se especificarán aquí dichos detalles. La única salvedad al analizar un video es que se agrega el segmento de tiempo en donde se identifica el contenido inapropiado.

2.2.1.3. Cloud Speech API

Esta herramienta tiene como objetivo la conversión de audio a texto. Sus principales características se especifican a continuación:

- Reconoce hasta 120 idiomas.
- Puede filtrar contenido inapropiado en los resultados para ciertos idiomas.
- Puntúa las transcripciones. Es decir, agrega comas, signos de interrogación y puntos.
- Transcribe en tiempo real o a través de audio ya grabado.
- Maneja el audio con ruido de fondo automáticamente, es decir, no requiere cancelación de ruido adicional.
- Reconoce nombres propios y logra dar formato a fechas, números o teléfonos.
- Retorna el instante de tiempo en el que se reconoce una frase.

2.2.1.4. Costos y financiación de las herramientas de Google

Un punto a destacar de Google es que ofrece un crédito de 300 USD gratuito para el uso de las herramientas por doce meses. Para ello, es necesario crear un usuario en Google Cloud Platform [21] y asociar una tarjeta de crédito, aunque no se efectuará ningún cargo hasta agotar el límite ya mencionado. Una vez pasado este punto, se debe abonar el uso de las herramientas en base a la cantidad de peticiones realizadas. Los costos se exponen al final de esta sección.

A modo de poder utilizar estas prestaciones durante el transcurso del proyecto, se apuntó a conseguir financiación. Se envió un correo dirigido al equipo comercial de Google en donde se

describió el contexto de estudio en el que se utilizarían sus herramientas y se solicitó crédito para hacer uso de ellas. Afortunadamente, se obtuvo una respuesta afirmativa y se logró conseguir el financiamiento solicitado.

Función	Precio por cada 1000 unidades		
	Primeras 1000 unidades al mes	De 1001 a 5.000.000 de unidades al mes	De 5.000.0001 a 20.000.000 de unidades al mes
Detección de etiquetas	Gratis	1,50 $	1,00 $
Detección de texto	Gratis	1,50 $	0,60 $
Detección de texto en documentos	Gratis	1,50 $	0,60 $
Búsqueda Segura (detección de contenido explícito)	Gratis	Gratis con Detección de etiquetas, o bien 1,50 $	Gratis con Detección de etiquetas, o bien 0,60 $
Detección facial	Gratis	1,50 $	0,60 $
Detección de puntos de referencia	Gratis	1,50 $	0,60 $
Detección de logotipos	Gratis	1,50 $	0,60 $
Propiedades de la imagen	Gratis	1,50 $	0,60 $
Sugerencias de recorte	Gratis	Gratis con Propiedades de la imagen, o bien 1,50 $	Gratis con Propiedades de la imagen, o bien 0,60 $
Detección web	Gratis	3,50 $	Ponte en contacto con nosotros para obtener más información

Figura 2.8: Costos de *Vision API* [45]

CARACTERÍSTICA	PRIMEROS 1000 MINUTOS	MINUTOS 1001-100,000
Detección de etiquetas	Gratis	$ 0.10 / minuto
Detección de disparo	Gratis	$ 0.05 / minuto, o gratis con detección de etiqueta
Detección SafeSearch	Gratis	$ 0.10 / minuto

Figura 2.9: Costos de *Cloud Video Intelligence API* [44]

Función	Hasta 60 minutos	Más de 60 minutos, hasta 1 millón de minutos
Reconocimiento de voz (todos los modelos excepto vídeo)	Gratis	0,006 $ cada 15 segundos*
Reconocimiento de voz en vídeo	Gratis	0,012 $ cada 15 segundos*

Figura 2.10: Costos de *Cloud Speech API* [43]

2.2.2. Amazon Rekognition

Amazon Rekognition es la herramienta desarrollada por Amazon que permite efectuar análisis de imágenes y videos. Expone una API simple y fácil de usar que procesa cualquier imagen o archivo de video almacenado en Amazon S3. Los servicios ofrecidos se clasifican en dos categorías: aquellos que permiten analizar videos (Video Features) y los que permiten analizar imágenes (Image Features).

2.2.2.1. Rekognition Image Features

Rekognition Image Features es el servicio de reconocimiento de imágenes que detecta objetos, escenas y rostros. A su vez, permite extraer texto e identificar celebridades o contenido inapropiado.

Detección de etiquetas

Esta herramienta detecta automáticamente objetos, conceptos y escenas. A su vez, para cada elemento, se proporciona un porcentaje de confianza con el cual fue identificado.

A continuación se presenta el resultado retornado para una imagen de ejemplo. En la figura 2.11 se observa la imagen analizada, mientras que el cuadro 2.1 recopila las etiquetas detectadas.

Según la respuesta obtenida, se detectaron cinco elementos: patineta (skateboard), deporte (sport), deportes (sports), humano (human) y persona (person). A su vez, se observa el porcentaje de confianza con el cual se reconoció cada una.

Figura 2.11: Imagen analizada por *Rekognition Image Features* para detección de etiquetas [2]

Etiqueta	Confianza
Skateboard	99,25
Sport	99,25
Sports	99,25
Human	99,24
Person	99,24

Cuadro 2.1: Etiquetas reconocidas por *Rekognition Image Features*

Detección de rostros

Otra de las funcionalidades ofrecidas por *Rekognition Image Features* es la detección de rostros. La respuesta de esta operación devuelve las siguientes propiedades para cada rostro detectado, junto con el porcentaje de confianza asociado.

- *Bounding box*: coordenadas del cuadro delimitador que rodea el rostro.
- *Puntos de referencia faciales*: coodenadas dentro de la imagen para cada punto de referencia detectado (ojo izquierdo, ojo derecho, boca, etc).
- *Atributos faciales*: atributos propios del rostro identificado. Estos pueden ser el sexo, rango de edad, o si la persona tiene barba o usa lentes.
- *Calidad*: describe el brillo y la nitidez de la cara.
- *Pose*: rotación del rostro dentro de la imagen.
- *Emociones*: emociones que logra identificar a través de las expresiones faciales. Estas pueden ser, por ejemplo, “alegría”, “tristeza” o “miedo”.

Debido a la longitud de la respuesta, se incluye en el apéndice A la información obtenida para el análisis de una imagen utilizando esta funcionalidad.

Detección de celebridades

A través de esta herramienta, *Rekognition Image Features* permite identificar celebridades en las imágenes analizadas. Puede detectar personalidades en una amplia gama de categorías, como entretenimiento, medios, deportes, negocios o hasta política. Reconoce hasta cien celebridades en una misma imagen e incluso devuelve enlaces a páginas web con información sobre las mismas.

La respuesta de esta funcionalidad incluye lo siguiente:

- *Celebridades reconocidas*: retorna una lista con los nombres de todas las celebridades detectadas. Además, incluye las URLs que direccionan a contenido relacionado con esa celebridad. Retorna también las coordenadas del rostro encontrado, así como algunos de sus atributos faciales.
- *Rostros no reconocidas*: retorna una lista con los rostros que no pudo reconocer, también junto a sus coordenadas y algunos atributos faciales.

A modo de ejemplo, se observa el siguiente extracto de la información obtenida a partir del análisis de la figura 2.12.

- URLs: [www.imdb.com/name/nm0461498]
- Name: Beyonce
- Puntos de referencia faciales:
    1. Ojo Izquierdo
        - "X": 0.5185439586639404
        - "Y": 0.2069191038608551
    2. Ojo Derecho
        - "X": 0.5823431015014648
        - "Y": 0.20255666971206665
    3. Nariz
        - "X": 0.5465836524963379
        - "Y": 0.23379606008529663
- Confianza: 99.99
- Rostros no reconocidos: Ninguno

Figura 2.12: Imagen para la detección de celebridades de *Rekognition Image Features* [23]

Moderación de imágenes

Esta funcionalidad tiene como objetivo la detección de material para adultos en las imágenes analizadas. La herramienta cataloga el contenido de la imagen dentro de un conjunto de categorías definidas, asociándole a cada detección un nivel de confianza. Se puede observar un ejemplo del resultado obtenido para la figura 2.13 a continuación:

Figura 2.13: Imagen para moderación de imágenes de *Rekognition Image Features* [31]

Categoría	Confianza
Sugestiva	98,75
Mujer en ropa interior o bikini	98,75

Cuadro 2.2: Resultados de moderación de imágenes de *Rekognition Image Features*

Detección de textos en imágenes

La última herramienta de *Rekognition Image Features* a presentar es la de reconocimiento y extracción de texto contenido en imágenes. Esta permite detectar tanto palabras sueltas como líneas. En este contexto, una palabra corresponde a uno o más caracteres que no están separados por espacios. Por otro lado, se entiende por línea a una cadena de palabras igualmente espaciadas, no necesariamente siendo una oración completa. Una línea finaliza cuando hay una gran brecha entre palabras en relación a la longitud de las mismas. A su vez, esta herramienta también puede detectar números y símbolos comunes.

Para la figura 2.14 se logra extraer la siguiente información:

- Texto Detectado: Spotify
- Confianza: 99.89
- Tipo: Palabra, Línea

Figura 2.14: Imagen analizada por *Rekognition Image Features*, detección de texto [37]

2.2.2.2. Rekognition Video Features

Esta herramienta cuenta con funcionalidades análogas a las de *Rekognition Image Features*, pero para la extracción de información en videos. Estas se listan a continuación:

Detección de etiquetas en videos

Su funcionamiento es equivalente a la herramienta para imágenes. La diferencia se da en que para los videos se detecta, además, el instante de tiempo en el que la entidad aparece.

Detección de rostros

Análoga a la herramienta para imágenes ya estudiada. La respuesta del análisis para videos agrega el instante del tiempo en donde se identificó el rostro. Cabe destacar que si el mismo aparece más de una vez, este será detectado en cada aparición.

Detección de celebridades

Esta funcionalidad se corresponde con la detección de celebridades en imágenes. La respuesta en este caso incluye todas las personalidades reconocidas y el momento en el que son detectadas.

Detección de contenido inapropiado

El cometido de esta herramienta es el mismo que su equivalente para imágenes. Sin embargo, al igual que para todas las funcionalidades de *Rekognition Video Features* se agrega en la respuesta el instante de tiempo en el que se detecta el contenido.

2.2.2.3. Costo y financiación de las herramientas de Amazon Rekognition

Con *Amazon Rekognition*, el costo se establece en base a la cantidad de imágenes o minutos de video analizados. En general, todas los servicios ofrecidos, salvo ciertas excepciones, no tienen costo hasta un límite determinado. Este se mide de manera diferente dependiendo de la herramienta en cuestión. Por ejemplo, en el caso de imágenes se pueden analizar 5.000 al mes durante los primeros 12 meses. Mientras que para videos, el límite se establece en 1.000 minutos por mes durante el primer año. Más abajo se pueden observar las tablas con los costos desglosados por funcionalidad.[2]

Al igual que con Google, se recurrió a un procedimiento para obtener financiación y así poder utilizar los sevicios *Amazon Rekognition*. Este proveedor cuenta con un plan que ofrece créditos a proyectos de investigación, por lo que se procedió con la postulación a dicho plan a través de su página web. [10]

EE.UU. ESTE (NORTE DE VIRGINIA)

Capas de análisis de imágenes	Precio por 1 000 imágenes procesadas
Primer millón de imágenes procesadas* al mes	1,00 USD
Siguientes 9 millones de imágenes procesadas* al mes	0,80 USD
Siguientes 90 millones de imágenes procesadas* al mes	0,60 USD
Más de 100 millones de imágenes procesadas* al mes	0,40 USD

* Cada API que acepta una o más imágenes de entrada cuenta como una imagen procesada. Más información »

Figura 2.15: Costos de *Rekognition Image Features* [46]

[2]Los costos varían por región. Se toma como referencia la más económica.

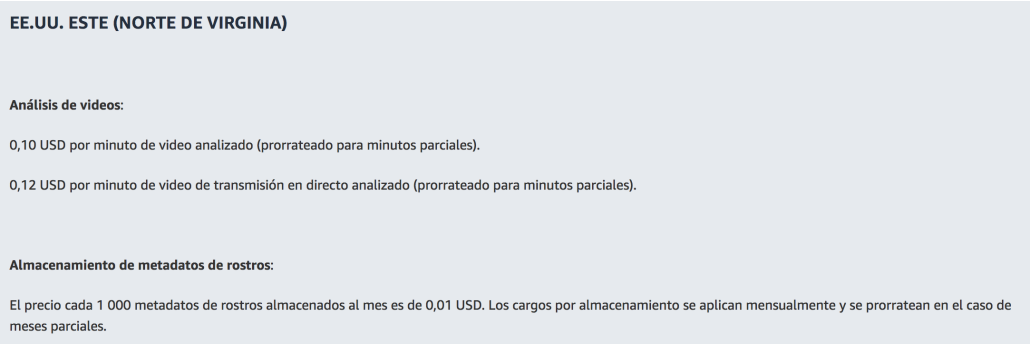


Figura 2.16: Costos de *Rekognition Video Features* [46]

2.2.3. Clarifai

Clarifai es una compañía dedicada al desarrollo de software de inteligencia artificial, especializada mayoritariamente en reconocimiento visual. Su herramienta *Predict*, tiene como objetivo el análisis de contenido multimedia para la extracción de información. Esta recibe como entrada un archivo de imagen o video y retorna una predicción sobre su contenido. Más específicamente, devuelve una lista de entidades encontradas junto con los valores de confianza.

Al ejecutar un pedido de reconocimiento, se debe especificar un modelo de datos a utilizar. Este contiene un grupo de conceptos relacionados y la herramienta solo podrá reconocer aquellos que se contienen en él. *Predict* cuenta con un grupo de modelos ya entrenados que están disponibles en su página web. [42] Entre ellos, dispone de uno "genérico" para uso universal. A su vez, es posible que el usuario cree y entrene su propio modelo adaptado a sus necesidades particulares.

A continuación se presentan las dos funcionalidades provistas por *Predict*: *Predict Images* y *Predict Videos*.

2.2.3.1. Predict Images

Para obtener predicciones para una imagen, esta debe estar disponible a través de una URL pública o ser enviada en forma de bytes en la solicitud realizada.

A continuación, se presenta un cuadro en donde se muestran algunos de los conceptos retornados para la imagen de la figura 2.17.

Id	Nombre	Valor	App_Id
ai_HLmqFqBf	Tren	0.9989112	main
ai_fvlBqXZR	Ferrocarril	0.9975532	main
ai_Xxjc3MhT	Sistema de transporte	0.9959158	main
ai_6kTjGfF6	Estación	0.992573	main
...	...	...	...

Cuadro 2.3: Algunos conceptos retornados por *Predict Images*

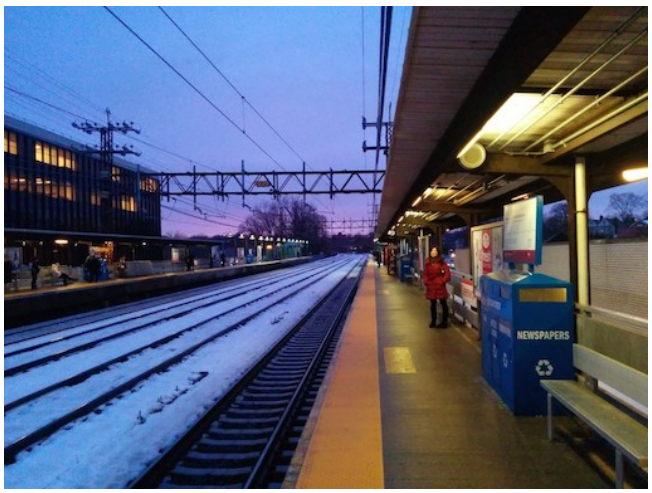


Figura 2.17: Imagen analizada por *Predict Images* [7]

2.2.3.2. Predict Videos

Esta herramienta es análoga a la anterior, la diferencia radica en que esta recibe videos como entrada. *Predict Videos* toma capturas del video cada un segundo y analiza cada una de estas. De esta forma, se obtiene una lista de conceptos por cada segundo de video.

En la figura 2.18 se observa un fragmento de la respuesta de *Predict Videos* para el primer segundo de un video de ejemplo.[3] El campo `time` especifica el tiempo (en milisegundos) al que corresponde el cuadro, mientras que `concepts` agrupa en una lista los conceptos encontrados para ese instante.


```
{
  "frame_info": {
    "index": 1,
    "time": 1000
  },
  "data": {
    "concepts": [
      {
        "id": "ai_zJx6RbxW",
        "name": "drink",
        "value": 0.98658466,
        "app_id": "main"
      },
      {
        "id": "ai_mCpQg89c",
        "name": "glass",
        "value": 0.97975093,
        "app_id": "main"
      },
```


Figura 2.18: Ejemplo de respuesta de *Predict Videos*

[3]Ejemplo extraído de la documentación oficial de Clarifai. `https://www.clarifai.com/developer/guide/predict#videos`

2.2.3.3. Costos y financiación de las herramientas de Clarifai Predict

En el caso de Clarifai, el sistema de costos se divide en cinco planes.

- *Unverified Email Plan*
  - Permite solo 100 pedidos por mes.
  - No tiene costo.
- *Community Plan*
  - 5.000 operaciones para usar por mes.
  - 10 conceptos personalizados gratuitos para usar en el entrenamiento personalizado.
  - 10.000 imágenes para almacenar en la nube.
  - No tiene costo.
- *Essential Plan*
  - Se abona por uso, con un límite de 100.000 operaciones.
  - Tiene un costo de 1,2 USD cada 1.000 operaciones.
  - 10.000 imágenes para almacenar en la nube.
- *Business Plan*
  - Ofrece descuentos por volumen de operaciones.
  - Se abona un precio fijo mensual.
  - Tiene como límite 1 millón de operaciones.
  - El plan comienza en 2.000 USD mensuales.
- *Enterprise Plan*
  - Adecuado para contextos en donde se requiere un gran volumen de operaciones por mes (más de 1 millón).

Para que el uso de esta herramienta en el proyecto sea posible, se recurrió una vez más a la búsqueda de financiación. Finalmente, se acordó que Clarifai brindaría un crédito de 500 USD para el libre uso de su tecnología.

2.2.4. Microsoft

2.2.4.1. Video Indexer

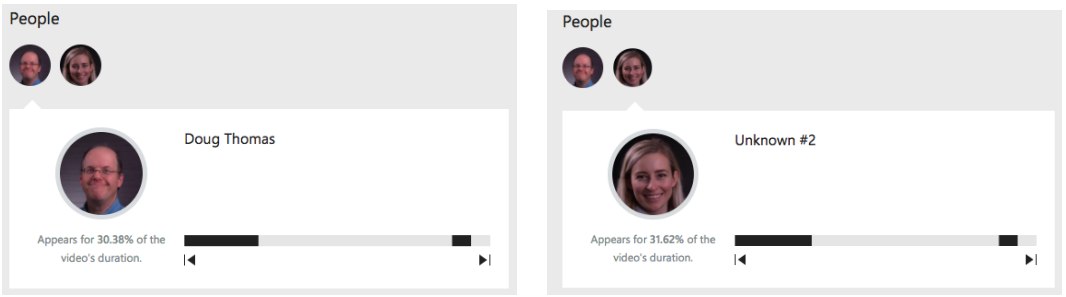
Video Indexer forma parte de los servicios ofrecidos por *Microsoft Cognitive Services*. Estos utilizan algoritmos de inteligencia artificial para ayudar a ver, oír, hablar, comprender e interpretar las necesidades de los usuarios a través de formas de comunicación naturales. A continuación se describen algunas de las funcionalidades provistas por esta herramienta.[4]

Transcripción de audio

Esta funcionalidad permite transformar audio a texto, incluyendo el instante de tiempo en el que se encuentran las frases detectadas. Los idiomas soportados incluyen Inglés, Español, Francés, Alemán, entre otros.

Reconocimiento e identificación facial

Hace posible la detección de rostros en un video. Las caras identificadas se comparan con una base de datos de personalidades para evaluar cuáles se encuentran en el video. Asimismo, los usuarios pueden crear etiquetas para rostros que no sean reconocidos. *Video Indexer* se encarga de crear un modelo basado en dichas etiquetas para poder reconocerlas en videos analizados en el futuro.



(a) Detección de personalidad conocida

(b) Detección de persona desconocida

Figura 2.19: Reconocimiento facial de *Video Indexer*

[4]Todos los ejemplos sobre *Video Indexer* se encuentran en las demos proporcionadas en su documentación oficial. `https://www.videoindexer.ai/?tab=samples`

Indexado de locutores

Tiene la capacidad de reconocer quién habló, qué palabras dijo y en qué instante de tiempo. Un ejemplo de esto se observa a continuación:

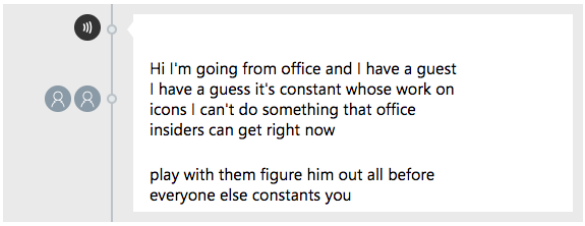


Figura 2.20: Locutores reconocidos por *Video Indexer*

Detección de escenas

Tiene la capacidad de realizar un análisis visual en el video y así determinar cuándo ocurre un cambio de escena.

Análisis de sentimiento

Realiza análisis de sentimientos sobre el texto extraído del video. Luego, retorna dicha información en forma de sentimientos positivos, negativos o neutros, junto con los instantes de tiempo en donde fueron encontrados.

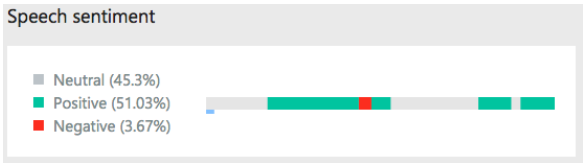


Figura 2.21: Análisis de sentimientos por *Video Indexer*

Detección de logos

Esta funcionalidad permite identificar logos o marcas reconocidas.

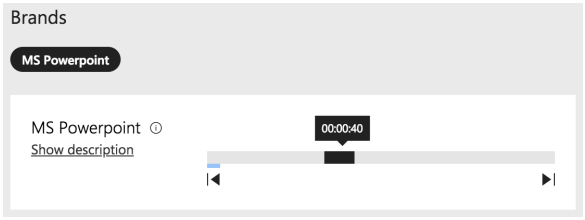


Figura 2.22: Detección de marcas por *Video Indexer*

2.2.4.2. Costos y financiación de las herramientas de Video Indexer

	Análisis de video	Análisis de audio
Precio por minuto de entrada	0,067 USD	0,016 USD

Cuadro 2.4: Costos por uso de *Video Indexer*

Al igual que para los anteriores proveedores, se solicitó financiación por parte de Microsoft de manera de poder utilizar sus servicios. Desafortunadamente, no se obtuvo una respuesta afirmativa, razón por la cual no se trabajará con esta herramienta en la solución desarrollada.

2.2.5. IBM

2.2.5.1. Watson Visual Recognition

A través de *Watson Visual Recognition*, IBM brinda servicios de reconocimiento visual para el análisis de imágenes, detectando objetos, rostros y otros contenidos.

Cuenta con diversos modelos ya incorporados que ayudan a guiar los resultados para el análisis de multimedia. A su vez, los usuarios pueden crear y entrenar sus propios modelos, adaptando así la herramienta a sus necesidades.

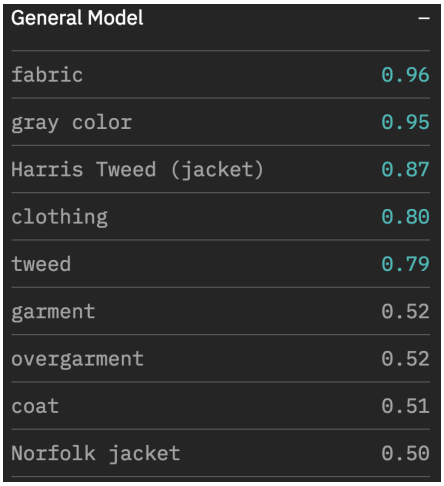
A continuación listan los modelos integrados:[5]

- *Modelo general*: clasificación predeterminada de diversos objetos, acciones o escenas.

  En el ejemplo representado en la figura 2.23, es posible observar que para cada elemento detectado se devuelve también un índice de confianza. Esto aplica para todos los modelos.

[5]Todos los ejemplos de análisis de *Watson Visual Recognition* fueron extraídos de la demo proporcionada por la documentación oficial. `https://www.ibm.com/watson/services/visual-recognition/demo/#demo`

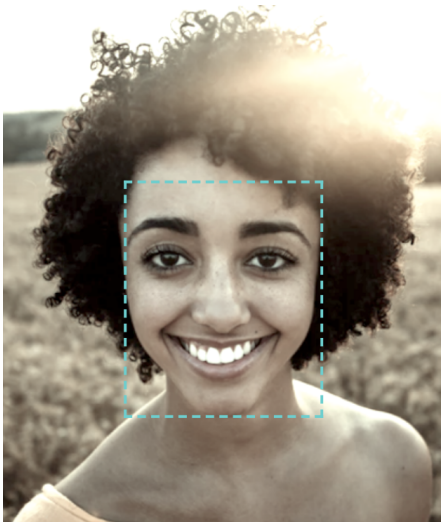
(a) Imagen seleccionada para el análisis



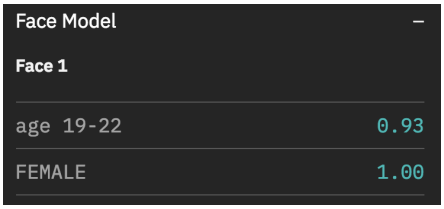
(b) Resultado obtenido

Figura 2.23: Análisis de *Watson Visual Recognition* utilizando el modelo general

- *Modelo facial*: para la detección de rostros y análisis de edad y sexo, entre otras características.



(a) Imagen seleccionada para el análisis

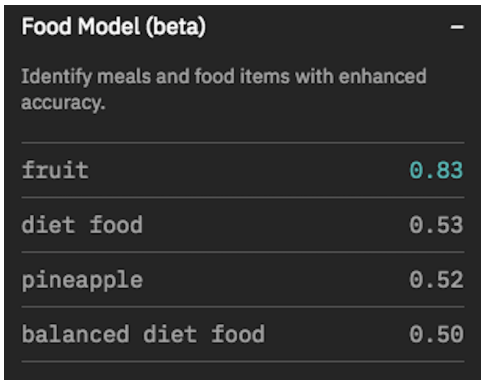


(b) Resultado obtenido

Figura 2.24: Análisis de *Watson Visual Recognition* utilizando el modelo facial

- *Modelo explícito*: determina si una imagen posee contenido inapropiado. Esta funcionalidad se encuentra en versión Beta.
- *Modelo de alimentos*: para utilizar específicamente en el análisis de imágenes que contengan alimentos. Esta funcionalidad se encuentra en versión Beta.

(a) Imagen seleccionada para el análisis

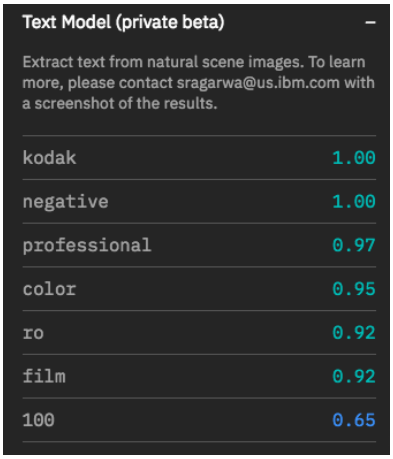


(b) Resultado obtenido

Figura 2.25: Análisis de *Watson Visual Recognition* utilizando el modelo de alimentos

- *Modelo de texto*: utilizado para reconocer texto presente en imágenes. Esta funcionalidad se encuentra en versión Beta.

(a) Imagen seleccionada para el análisis



(b) Resultado del análisis para el modelo de texto

Figura 2.26: Análisis de Watson Visual Recognition, modelo de texto

2.2.5.2. Costos y financiación de las herramientas de Watson Visual Recognition

Watson Visual Recognition ofrece un período de prueba gratuito que abarca el análisis de 1000 imágenes, usando tanto modelos incorporados como personalizados.

Una vez superado este límite, los costos serían los siguientes:

- Clasificación de imágenes: 0,002 USD por imagen.
- Detección de rostros: 0,004 USD por imagen.

Se buscó conseguir financiación de parte de IBM a modo de utilizar los servicios ofrecidos. Sin embargo, no se obtuvo respuesta, por lo que Watson Visual Recognition no podrá ser incluida en el desarrollo de este proyecto.

2.2.6. Resumen comparativo de las herramientas estudiadas

A partir de la anterior indagación, es posible concluir que existen diversas herramientas para el análisis de imágenes, audio y video que pueden colaborar para alcanzar los objetivos buscados en este proyecto. A modo de resumen, se presenta la tabla 2.5 en donde se observan las funcionalidades provistas por cada una de ellas.

De aquí se desprende que las herramientas existentes ofrecen un conjunto de funcionalidades diferentes para aportar valor a sus respuestas. Se puede ir desde la más básica como *Clarifai Predict* con sus detectores de etiquetas, pasando por los detectores de texto de *Amazon Rekognition* y *Google Vision API* que aprovechan casi todas las oportunidades de detección de texto. El reconocedor de voz de *Cloud Speech API* es extremadamente preciso y transcribe un gran porcentaje de las palabras existentes en un diálogo. También se cuenta con el detector de celebridades de *Amazon Rekognition* que tanto en imágenes como videos logra reconocer personalidades famosas correctamente. En tanto a las herramientas que no pueden ser utilizadas por no contar con crédito disponible, más allá de sus pruebas gratuitas, *Watson Visual Recognition* de IBM no aporta características diferentes a las que se pueden obtener con las herramientas de procesamiento de imágenes de Google y Amazon. *Video Indexer* de Microsoft parece ser la más completa, de todos modos, contiene funcionalidades que a los efectos de construir un índice no son tan excluyentes, como la identificación de locutores no conocidos dentro de una conversación. Además, sus funcionalidades más potentes, como lo son el reconocimiento de celebridades y el de audio, ya están cubiertas por otras de las herramientas estudiadas.

Por otro lado, hay que destacar dos puntos. El primero, es que la única herramienta que soporta el reconocimiento de lugares de interés es *Vision API*. Dicha funcionalidad se considera fundamental para identificar entidades tales como la Torre Eiffel o el Coliseo Romano.

El segundo punto a resaltar, es que todas los proveedores ofrecen la funcionalidad de detección de etiquetas. Esto no quiere decir que produzcan la misma información, ya que los modelos utilizados por cada servicio pueden variar, provocando que se reconozcan elementos distintos. De esta manera, si bien la funcionalidad es la misma, las respuestas obtenidas podrían complementarse entre sí para brindar mejores resultados.

Finalmente, es posible concluir que cada herramienta puede aportar una pieza de información que sirva para armar el rompecabezas que se busca resolver: generar índices sobre videos. Tal vez por separado no alcancen para ubicar los momentos más importantes, pero haciendo que trabajen en conjunto, se puede aprovechar lo mejor de cada una de manera de obtener la información necesaria para alcanzar el objetivo.

	Google			Amazon		Clarifai		Microsoft	IBM
	Vision API	Cloud Video Intelligence	Cloud Speech API	Rekognition Image Features	Rekognition Video Features	Predict Images	Predict Videos	Video Indexer	Watson Visual Recognition
Detección de etiquetas	✓	✓	N/A	✓	✓	✓	✓	✓	✓
Detección de rostros	✓	✓	N/A	✓	✓	✗	✗	✓	✓
Detección de contenido explícito	✓	✓	N/A	✓	✓	✗	✗	✓	✓
Detección de celebridades	✗	✗	N/A	✓	✓	✗	✗	✓	✗
Detección de puntos de referencia	✓	✗	N/A	✗	✗	✗	✗	✗	✗
Detección de logos	✓	✗	N/A	✗	✗	✗	✗	✓	✗
Detección de texto	✓	✗	N/A	✓	✗	✗	✗	✓	✓
Detección de cambios de escenas	N/A	✗	N/A	N/A	✗	N/A	✗	✓	✗
Detección de locutores	N/A	✗	✗	N/A	✗	N/A	✗	✓	N/A
Transcripción de audio	N/A	✗	✓	N/A	✗	N/A	✗	✓	N/A
Financiación	✓			✗		✓		✗	✗

Cuadro 2.5: Comparación de funcionalidades de herramientas

Capítulo 3

Pruebas de concepto

El objetivo de este capítulo consta de expresar los resultados del análisis realizado acerca de las herramientas descritas en el capítulo anterior. Esto llevará a poder determinar cuáles son las que realmente aportarán valor a la solución y cuáles podrán ser desestimadas.[1]

Para las pruebas, se seleccionan diferentes videos, imágenes y audios y se procesan con las herramientas correspondientes, registrando los resultados obtenidos.

Finalmente, se comparan las respuestas diferenciando entre imágenes, audio o videos, y se determinan qué herramientas resultan ser aplicables al proyecto. La evaluación de cada una se realiza siguiendo los criterios especificados en la siguiente sección.

3.1. Criterios para la evaluación

- *Falsos positivos*: se refiere a que la herramienta devuelva lo que efectivamente se encuentra dentro del contenido. Es decir, no se deberían retornar entidades que no se correspondan con lo que se observa o escucha.

- *Grado de detalle*: se refiere a qué tan específicos son los resultados obtenidos. A modo de ejemplo, en una imagen donde se observa una rosa, una respuesta que identifique "flor" se considera menos acertada que otra en donde se identifique el tipo de flor encontrada.

- *Elementos no detectados*: se refiere a aquellos elementos, lugares o personas que no fueron reconocidos. Un ejemplo podría ser una imagen donde se observa un gato y un perro pero la herramienta solo logra detectar uno de ellos.

- *Veracidad del nivel de confianza*: se refiere a que se corresponda el nivel de confianza retornado por la herramienta con lo que se puede ver o escuchar en el video. En otras palabras, se desea que si el nivel de confianza es alto, la detección sea correcta y si es medio o bajo, sea probablemente una detección errónea.

[1] El análisis se realizará teniendo en cuenta únicamente las herramientas para las cuales se obtuvieron respuestas en las solicitudes de financiamiento.

- *Relevancia dentro de contexto*: se refiere a qué tan significativa es para el video la información obtenida. Por ejemplo, en un video sobre una receta de cocina, sería más relevante reconocer los ingredientes de la misma, que detectar objetos ubicados en el fondo de la escena.

3.2. Conclusiones de la evaluación

A partir de las pruebas realizadas, se desprende una clasificación de las herramientas en tres conjuntos: aquellas que aportan valor para cualquier categoría de video, las que adquieren relevancia dentro de contextos particulares y las que terminaron siendo descartadas.

En el apéndice A, se pueden encontrar extractos de las respuestas obtenidas por las diferentes herramientas para alguno de los videos e imágenes seleccionados para la evaluación.

3.2.1. Herramientas de valor a nivel general

3.2.1.1. Detección de etiquetas

Por un lado, se concluye que la funcionalidad de detección de etiquetas (provista por los tres proveedores de servicios estudiados) puede ser aplicable a cualquier categoría de video o imagen. Es decir, esta funcionalidad puede reconocer elementos tanto en videos musicales como noticias o programas de naturaleza. De esta manera, logra aportar valor a la construcción de los índices a nivel general.

Por otro lado, se identificaron casos en donde se introducen falsos positivos, detectando etiquetas que no se corresponden con el contenido real. De todas formas, restringiendo los resultados a aquellos que superen cierto nivel de confianza, se puede mitigar dicho riesgo.

Finalmente, se observó que el grado de detalle de las detecciones es variado. En muchos casos, se reconocen elementos específicos, pero también se encuentran en las respuestas entidades muy generales. Ejemplos de etiquetas generales que suelen aparecer son, “animal”, “persona”, “lugar” u “horizontal”.

3.2.1.2. Detección de texto

En cuanto a los servicios que reconocen texto en imágenes, se puede concluir que son herramientas de gran potencial. Suelen detectar la mayoría de las cadenas de caracteres que aparecen en una figura, brindando información de gran valor en estos casos.

En contrapartida, muchas veces incluyen fallos al perder algunas letras o confundiéndolas con otras debido al tipo de fuente de texto utilizada. De todas formas, se estima que con un procesamiento adicional, se pueden explotar los beneficios de estas herramientas.

3.2.1.3. Reconocimiento de audio

El servicio de transformación de voz a texto que ofrece *Cloud Speech API* tiene un alto grado de acierto y detalle en las palabras reconocidas. Posee dos contras a la hora de la generación de índices. Una de ellas es que el valor de esta información se pierde en videos donde el audio no coincide con una secuencia de voz (por ejemplo, música instrumental). La segunda desventaja, es que se pueden introducir índices en momentos incorrectos. Un ejemplo de esto, es el caso de un relator de fútbol hablando sobre un gol que fue concretado en otro momento del partido. En este caso, se podría dar que el índice incluya el hito "gol" en un instante donde en realidad está ocurriendo otra cosa en el juego. Cabe destacar, que esto no se considera un falso positivo de la herramienta, ya que se estaría reconociendo la palabra de forma correcta.

3.2.2. Herramientas de valor en contextos particulares

3.2.2.1. Detección de logos

La detección de logos de *Vision API* adquiere relevancia únicamente en contextos donde este tipo de entidad aparece. Durante las pruebas con esta funcionalidad no se observaron falsos positivos. A su vez, suele reconocer todos los logos presentes en una imagen.

3.2.2.2. Detección de lugares de interés

Esta funcionalidad de *Vision API* es de gran utilidad en escenarios donde se muestran distintos sitios del planeta. No aporta gran valor en contenidos que se llevan a cabo en interiores.

A veces, su grado de confianza es muy bajo para detecciones correctas. Por ejemplo, reconoce correctamente "Bethesda Fountain" en Central Park con una confianza menor al 40 % (ver apéndice A.2.4), valor que puede ser considerado poco fiable. Esto podría llevar a la aparición de falsos positivos, ya que con el mismo nivel de confianza se pueden detectar lugares incorrectos, dificultando la distinción entre información verídica y falsa.

3.2.2.3. Reconocimiento de celebridades

Esta funcionalidad expuesta por *Rekognition Video Features* y *Rekognition Image Features* posee un alto grado de detalle. Su valor se ve claramente reflejado en videos donde hacen su aparición personalidades conocidas, teniendo como aspecto positivo que el conjunto de celebridades reconocidas es bastante amplio. En este sentido, se logran reconocer deportistas, cantantes y actores, pero también científicos, empresarios o políticos. (Ver apéndices A.1.2 y A.2.5)

Durante las pruebas se observó que se suelen reconocer la mayoría de las personalidades que figuran en la imagen o video. En general, las personas no identificadas se dan en casos donde su rostro se ve borroso o poco descubierto.

En tanto a los niveles de confianza, se percibió algo particular. Para detecciones erróneas, los porcentajes de confianza se consideran altos, ubicándose entre el 50 y 85 %. Mientras que todos los reconocimientos acertados superan el 90 o 95 %. Por lo tanto, se desprende la conclusión de

que se debe determinar un nivel de confianza mínimo aceptable del 90 % con el fin de mitigar el riesgo de introducir falsos positivos. Esta característica se ilustra, a su vez, en un curioso reporte donde, a través de los servicios de *Rekognition Video Features*, se detectaron coincidencias entre miembros del congreso estadounidense con 28 ladrones conocidos. [64]

3.2.3. Herramientas descartadas

3.2.3.1. Detección de rostros

Durante las pruebas, se observó que la funcionalidad de detección de rostros, tanto para imágenes como videos, no aportaban información de valor a la hora de generar un índice. Estas se enfocan más que nada en localizar físicamente los rostros dentro de una imagen y no en reconocer características relevantes para utilizar en los índices. Es por ello que estos servicios provistos por Amazon y Google no serán utilizados en el prototipo desarrollado. Ejemplos de los resultados obtenidos por esta herramienta se encuentran en el apéndice A.2.2.

3.2.3.2. Moderación de contenido explícito

Se decidieron excluir, también, las funcionalidades de moderación de contenido en imágenes y videos. Si bien la información provista en un principio podría considerarse relevante dentro de un índice, a partir de las pruebas realizadas se concluyó que, en la mayoría de los casos, las detecciones de contenido inapropiado eran incorrectas.

A su vez, los resultados obtenidos eran demasiado genéricos como para ser considerados hitos dentro de un índice. Por ejemplo, un resultado que detecte que "probablemente cierta imagen sea sugestiva", no aporta demasiado valor dentro del contexto del problema a resolver. Todo esto se ve reflejado en los ejemplos suministrados en el apéndice A.2.7.

3.2.3.3. Detección de etiquetas

Se observó que la herramienta de procesamiento de imágenes *Predict Images* de Clarifai suele devolver varios resultados erróneos con porcentaje de confianza alto (ver apéndice A.2.1). En consecuencia, al contar con otros detectores de etiquetas que proveen la misma funcionalidad, se decidió no utilizar esta herramienta para el desarrollo del prototipo.

3.2.4. Evaluación general

En el cuadro 3.1 se observa la evaluación general para cada una de las herramientas. La misma se realizó en base a la información obtenida a partir de las pruebas realizadas. A su vez, se remarcan aquellas que finalmente serán utilizadas en la construcción del prototipo.

			Falsos positivos	Grado de detalle	Elementos no detectados	Nivel de confianza	Relevancia
Herramientas para imágenes	Detección de etiquetas	Rekognition Image Features	Sí	Bajo	Sí	Medio	Media
		Vision API	Sí	Bajo	Sí	Medio	Media
	Detección de rostros	Rekognition Image Features	No	Alto	No	Alto	Baja
		Vision API	No	Alto	No	Alto	Baja
	Detección de texto	Rekognition Image Features	Sí	Alto	No	Alto	Alta
		Vision API	Sí	Alto	No	Alto	Alta
	Moderación de imágenes	Rekognition Image Features	Sí	Medio	Sí	Medio	Baja
		Vision API	Sí	Bajo	Sí	Medio	Baja
	Detección de celebridades	Rekognition Image Features	Sí	Alto	Sí	Alto	Alta
	Detección de lugares de interés	Vision API	No	Alto	No	Bajo	Alta
	Detección de logos	Vision API	No	Alto	No	Medio	Alta
Herramientas para videos	Detección de etiquetas	Rekognition Video Features	Sí	Bajo	Sí	Medio	Media
		Cloud Video Intelligence	Sí	Bajo	Sí	Medio	Media
		Predict Video	Sí	Bajo	Sí	Medio	Media
	Detección de rostros	Rekognition Video Features	No	Alto	No	Alto	Baja
	Detección de celebridades	Rekognition Video Features	Sí	Alto	Sí	Alto	Alta
	Moderación de videos	Rekognition Video Features	Sí	Medio	Sí	Medio	Baja
	Detección de cambio de escena	Cloud Video Intelligence	No	N/A	No	No tiene	Alta
Herramientas para audio	Detección de audio	Cloud Speech API	No	Alto	Sí	Alto	Alta

Cuadro 3.1: Resumen de resultados de las pruebas realizadas

Capítulo 4

Especificación funcional

En el capítulo anterior se desplegaron los resultados de las pruebas de concepto realizadas para decidir qué herramientas de procesamiento se utilizarían en la solución a desarrollar.

En el presente, se describe la especificación funcional de dicha solución. Se comienza delimitando su alcance e indicando cómo las anteriores herramientas ayudarán a cumplirlo. Seguidamente, se detallan los requerimientos funcionales y no funcionales que debe cumplir el sistema para lograr abordar la problemática planteada.

4.1. Alcance

El prototipo a construir incluirá el desarrollo de una aplicación back-end capaz de generar índices para videos de manera automática. El índice se conformará de etiquetas que pueden representar objetos, lugares, personas, acciones, entre otros. Cada una de ellas asociada a un instante de tiempo. De esta forma, al navegar en el video por los momentos determinados en el índice, se deberían encontrar las etiquetas reconocidas. Esto será alcanzado a través del uso de las herramientas de procesamiento de imágenes, video y audio seleccionadas anteriormente. En el prototipo se buscará que estas trabajen en conjunto brindando información acerca del contenido del video.

Otro punto a destacar, es que el back-end debe ser capaz de integrarse con Video++1. Para esto, el formato del índice retornado debe cumplir con el modelo especificado en dicho proyecto para que pueda ser consumido por su aplicación web. El formato en cuestión será expuesto más adelante, en la sección 4.3.5.

Adicionalmente, se plantea el desarrollo de una interfaz de administración (front-end) que facilite la visualización de los resultados generados por el back-end. Esta tendrá como objetivo proporcionar una interfaz amigable, que ayude a visualizar y comparar los índices obtenidos por los distintos servicios utilizados. Este componente será únicamente para uso interno con el objetivo de evaluar el desempeño de las tecnologías incorporadas.

4.2. Requerimientos funcionales

Aquí se especifican tanto los requerimientos previstos desde un inicio, así como también aquellos surgidos durante el transcurso del proyecto.

4.2.1. Requerimientos planificados

4.2.1.1. Generar índice para un video

Este requerimiento es la base fundamental del prototipo a construir. Se debe brindar la funcionalidad de generar un índice para un video determinado.

Nombre: Generar índice.
Objetivo: Generar el índice para un video solicitado.
Actores: Usuario.
Precondiciones: - El video debe estar disponible en una URL pública.
Descripción: El Usuario envía al Sistema una solicitud para generar el índice sobre un video. A partir de esto, el Sistema le otorga un identificador para dar seguimiento al procesamiento que se dispara simultáneamente.
Flujo normal: 1. El Usuario solicita la generación del índice para un video. 2. El Sistema responde al usuario con un identificador generado para el video y un estado de procesamiento. Comienza la creación del índice de manera asincrónica. 3. El Sistema descarga el video y lo sube a google-cloud-storage y Amazon S3. Además, extrae el audio y una serie de imágenes de cuadros del video. 4. El Sistema consulta las herramientas de procesamiento de audio, video e imágenes para obtener información sobre el video. 5. El Sistema procesa la información obtenida, normaliza el formato de los datos. 6. El Sistema filtra hitos que contengan palabras irrelevantes o inexistentes. Este paso busca depurar la información. 7. El Sistema genera el índice unificando la información obtenida por los servicios de procesamiento. 8. El Sistema almacena el índice generado para el video. 9. El Sistema notifica a Video++1 del fin del procesamiento, enviando el índice generado.
Flujos alternativos: 3A. El video ya había sido procesado por el Sistema de manera exitosa. 1. El Sistema retorna el índice ya almacenado sin realizar ningún procesamiento adicional. 4A. Falla la llamada a alguna de las herramientas de procesamiento. 1. El Sistema reintenta el llamado una vez. 2. Si falla, el Sistema registra que ocurrió un error en el análisis con dicha herramienta. 3. Continúa el paso 5.
Postcondiciones: - Queda almacenada la información acerca del procesamiento del video, además del índice generado.

4.2.1.2. Obtener un índice previamente generado

Este requerimiento se refiere a retornarle al usuario el índice previamente generado para un video en particular.

Nombre: Obtener índice.
Objetivo: Obtener el índice generado para el video solicitado.
Actores: Usuario.
Precondiciones: - El video fue previamente procesado por el Sistema de forma exitosa.
Descripción: El Usuario solicita el índice generado para un video. El Sistema retorna la información requerida.
Flujo normal: 1. El Usuario realiza la solicitud para obtener el índice de un video. 2. El Sistema calcula el identificador con el cual se almacenó el índice para ese video. 3. El Sistema utiliza el identificador para recuperar el índice almacenado. 4. El Sistema retorna al Usuario el índice almacenado para el video solicitado.
Postcondiciones: - No hay postcondiciones.

4.2.1.3. Conocer el estado de procesamiento de un video

Se devuelve el estado en el que se encuentra el procesamiento de un video. Este puede ser uno de los siguientes:

- `IN_PROGRESS`: el video sigue en procesamiento.
- `SUCCESS`: se finalizó el procesamiento con éxito.
- `CLIENT_FAILURE`: se finalizó el procesamiento pero alguna de las herramientas utilizadas falló.
- `FAILURE`: el procesamiento falló, no se pudo generar el índice.

Nombre: Obtener estado del procesamiento.
Objetivo: Obtener el estado en el que se encuentra el procesamiento de un video.
Actores: Usuario.
Precondiciones: - Se realizó previamente una solicitud de procesamiento del video.
Descripción: El Usuario solicita el estado del procesamiento del video pasando como parámetro la URL del mismo. El Sistema devuelve la información requerida.
Flujo normal: 1. El Usuario realiza el pedido del estado del procesamiento, pasando la URL como parámetro. 2. El Sistema calcula el identificador del video en base a la URL. 3. El Sistema utiliza el identificador para buscar en la base de datos la información requerida. 4. El Sistema devuelve el estado en el que se encuentra el procesamiento del video solicitado.
Postcondiciones: - No hay postcondiciones.

4.2.1.4. Interfaz de administración (front-end)

Se presenta una interfaz web para uso interno que permita la visualización de los índices producidos por el sistema. La misma debe ser capaz de mostrar la respuesta generada por cada una de las herramientas utilizadas y el estado de procesamiento en el que se encuentran. Estos requerimientos se pueden reflejar de la siguiente manera, dividiéndolos en los siguientes casos:

Login

Los usuarios deben estar previamente autenticados y autorizados para poder utilizar cualquier funcionalidad brindada por la interfaz de administración.

Generar índice

A través de la interfaz de administración es posible solicitar la generación de índices para videos. Esto desencadena la ejecución del requerimiento 4.2.1.1.

Historial de videos

Se despliega una lista con los videos existentes en el sistema. Para cada video se incluye su título, fecha de creación, fuente de procedencia y estado de procesamiento en el que se encuentra.

Reproductor de video

A modo de simplificar la validación del índice generado, se incluye un visualizador que permite reproducir el video seleccionado. Esta funcionalidad es compatible únicamente con videos provenientes de YouTube.

Visualizar la información generada para un video

Nombre: Obtener resumen general del video.
Objetivo: Centralizar la información de procesamiento de un video. Visualizar el índice generado y los estados de procesamiento de los proveedores.
Actores: Usuario.
Precondiciones: - Se realizó previamente una solicitud de procesamiento del video.
Descripción: El Usuario accede a la página del video. Encuentra información acerca del mismo, como su título o categoría a la que pertenece. Además, se despliegan los índices generados y el estado de procesamiento de cada proveedor.
Flujo normal: 1. El Usuario se autentica en el Sistema. 2. El Usuario se dirige a la sección de videos procesados. 3. El Usuario selecciona un video de la lista. 4. El Sistema muestra en pantalla información general del video, el estado de cada uno de los proveedores y si todos han terminado, el índice generado.
Postcondiciones: - No hay postcondiciones.

Nombre: Obtener índice por proveedor.
Objetivo: Mostrar el índice generado por cada uno de los proveedores.
Actores: Usuario.
Precondiciones: - Se realizó previamente una solicitud de procesamiento del video y el procesamiento del proveedor seleccionado ha finalizado correctamente.
Descripción: El Usuario accede a la página de uno de los videos. Luego, selecciona una de las herramientas de procesamiento de la lista y visualiza los resultados generados por la misma.
Flujo normal: 1. El Usuario se autentica en el Sistema. 2. El Usuario se dirige a la sección de videos procesados. 3. El Usuario selecciona un video de la lista. 4. El Usuario selecciona una herramienta de procesamiento. 5. El Sistema muestra en pantalla el índice generado la herramienta elegida para el video seleccionado.
Postcondiciones: - No hay postcondiciones.

4.2.2. Requerimientos no planificados

4.2.2.1. Hitos más relevantes por categoría

Este requerimiento surge una vez ya encaminado el desarrollo del prototipo, luego de una reunión entre los integrantes de Video++1 y Video++2 donde se da a conocer una funcionalidad interesante del primer proyecto.

En Video++1 se le permite a un usuario dar "me gusta" a los hitos que considere más interesantes dentro del índice generado para un video. En base a esto, se percibió que dichos datos también podrían resultar útiles para Video++2 de la siguiente manera:

Se tendría un listado de etiquetas por categoría, en donde se almacenaría la cantidad de veces que se le da "me gusta" desde Video++1 a dicha etiqueta para un video asociado a esa categoría. De esta manera, se obtendría una muestra de las entidades que más interesa identificar para una categoría en particular.

Esto permitiría que, para videos relacionados, se conozcan las etiquetas más populares abriendo un amplio abanico de posibilidades para explotar dicha información en favor de la construcción de índices que aporten mayor valor.

A continuación, se plantea la especificación sobre cómo se implementa este requerimiento dentro del contexto de Video++2.

Nombre: Incrementar "me gusta" de un hito
Objetivo: Aumentar en uno la cantidad de "me gusta's" asociados a un hito detectado en un video, dentro de la categoría a la que este pertenece.
Actores: Usuario.
Precondiciones: - No hay precondiciones.
Descripción: Desde Video++1, un usuario le da "me gusta" a un hito dentro del índice generado para un video. Esta información es enviada a Video++2, quien se encarga de aumentar la cantidad de "me gusta's" que ese hito tiene dentro de la categoría a la que pertenece el video.
Flujo normal: 1. El Sistema recibe el hito y URL del video para el cual el Usuario dio "me gusta" desde Video++1. 2. El Sistema obtiene la categoría asociada al video. 3. El Sistema aumenta la cantidad de "me gusta's" asociados a ese hito dentro de la categoría en cuestión.
Flujos alternativos: 2A. No se encuentra el video guardado en la base de datos. 1. Se consulta la API de Youtube para obtener la categoría del video. 2. Continúa el paso 3.
Postcondiciones: - En caso de ser el primer "me gusta" para el hito, se crea el registro en donde se almacenará la cantidad con valor uno. En caso de ya existir, se aumenta en una unidad el valor previamente almacenado.

4.3. Requerimientos no funcionales

4.3.1. Resiliencia

En el contexto de este desarrollo, la resiliencia implica que el sistema continúe funcionando a pesar de posibles fallas de las herramientas de procesamiento utilizadas. Si esto ocurre, el sistema igualmente debe ser capaz de generar el índice para el video, aunque no cuente con toda la información.

4.3.2. Debe correr en la nube

El sistema debe ser alojado en la nube de manera que pueda ser accedido por otras aplicaciones, en particular por Video++1.

4.3.3. Extensibilidad

Debido al ritmo de desarrollo en tecnologías de procesamiento de imágenes, audio y video, es importante prever el crecimiento que pueda tener el sistema a futuro. Este requerimiento implica que sea sencillo poder agregar nuevos servicios o herramientas que se integren con la aplicación. No se debería necesitar un rediseño en la arquitectura para poder introducir nuevos componentes. Los detalles de cómo agregar una nueva herramienta de procesamiento se pueden consultar en el apéndice D.

4.3.4. Minimización de costos de procesamiento

Los recursos de procesamiento utilizados tienen un costo monetario. Es de gran interés poder minimizar dicho costo con el fin de ahorrar, no solo recursos, sino que también dinero. Esto comprende el uso de los servicios de la forma más eficiente posible.

4.3.5. Interoperabilidad

El sistema debe ser capaz de integrarse con la plataforma Video++1. El formato de los índices retornados debe seguir los lineamientos establecidos para que ambos proyectos puedan funcionar en conjunto.

Este formato se define como un XML de la siguiente manera:

```
<video_indexes>
    <data>
        <text start="3.3" dur="2.1">Puesta de sol</text>
        ...
    </data>
</video_indexes>
```

En donde `start` y `dur` indican el instante de tiempo en el que comienza el hito y su duración respectivamente (en segundos).

4.3.6. Tamaño y duración del video a analizar

El sistema tiene como restricción que los videos no superen los 30 minutos. Una vez que se pasa este umbral, no se garantiza el correcto procesamiento. Esta limitación surge a partir de que las herramientas utilizadas fallan la mayoría de las veces que se las consulta con videos de tal duración. A su vez, debido a limitaciones propias del servicio de Clarifai, un video que supere los 80MB o 10 minutos de duración no será procesado por dicha herramienta.

Por otro lado, videos que no provengan de Youtube no serán procesados por la herramienta *Cloud Speech API* debido a problemas en la conversión del formato de los archivos.

4.3.7. Idiomas soportados

El sistema soporta dos idiomas: inglés y español. En caso de que se quiera procesar un video que no pertenezca a los idiomas soportados, los resultados obtenidos serán menos precisos.

Capítulo 5

Diseño de la solución

En el presente capítulo se describen los componentes necesarios y las interacciones entre los mismos para concretar los requerimientos especificados en el capítulo anterior.

5.1. Arquitectura del sistema

El sistema consta de tres componentes principales: un back-end (responsable de la lógica de negocio), una base de datos (encargada del almacenamiento de la información) y un front-end (interfaz web de uso interno). En la figura 5.1 se representa el diagrama de arquitectura, en donde se muestra la interacción entre ellos. Se indica, a su vez, los protocolos de comunicación utilizados entre los módulos y los servicios externos consumidos.

Dicho diagrama será utilizado a lo largo del capítulo para identificar cómo cada componente participa en la resolución de los diferentes problemas abordados.

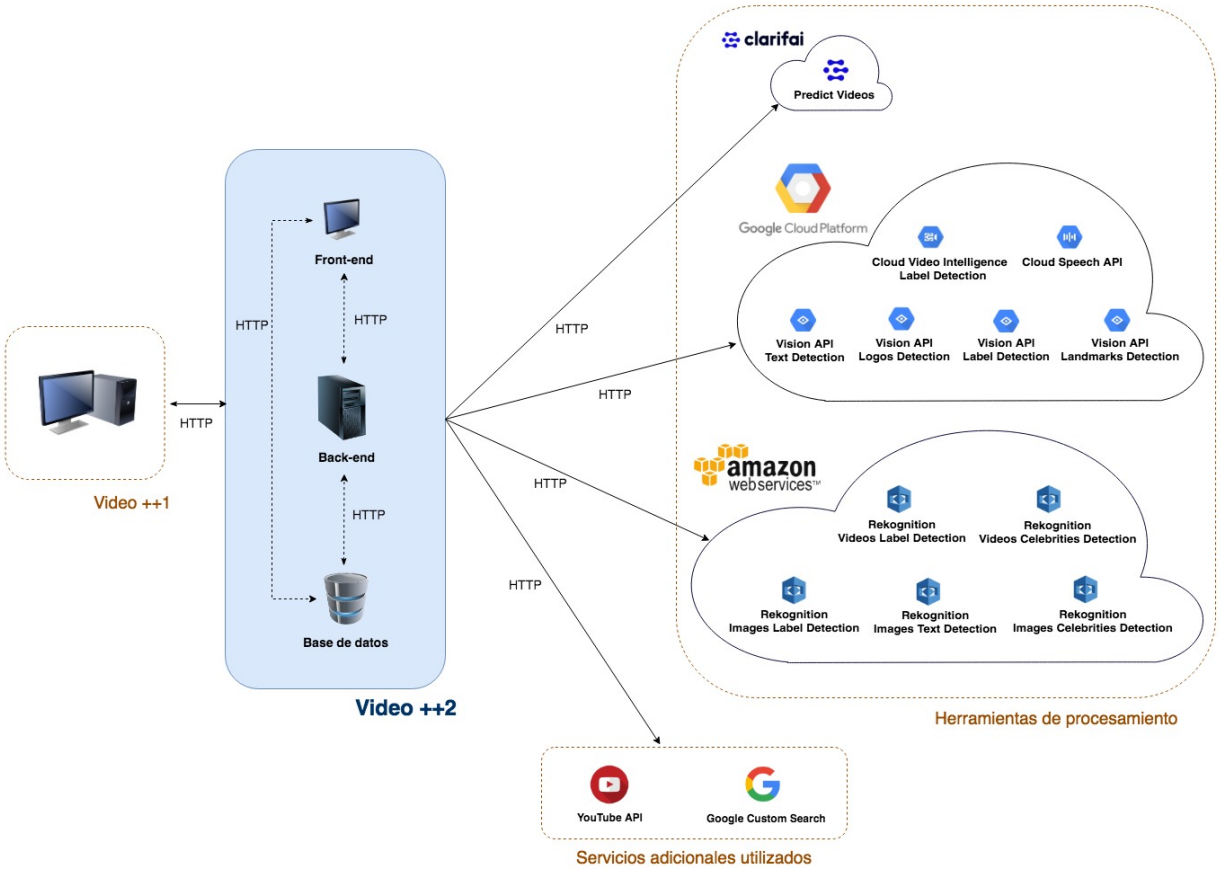


Figura 5.1: Diagrama de arquitectura del sistema

5.2. Back-end

Este componente ataca el problema principal del proyecto, la creación de índices de forma automática para videos. En tal sentido, se dividió el problema en tres: la generación de información relacionada a un video, la normalización de dicha información, y finalmente, el filtrado y unificación de la misma. A lo largo de esta sección se describen los mecanismos diseñados para encarar los anteriores problemas.

Por último, se expone una estrategia ideada con el fin de obtener otros datos para mejorar, en un futuro, la calidad de los índices.

5.2.1. Generación de la información

Para la construcción de los índices, se hace un análisis tanto del video, como del audio e imágenes obtenidas a partir del mismo. De esta manera, se puede sacar provecho de una mayor cantidad de herramientas de procesamiento.

5.2.1.1. Análisis de video

Implica el uso de herramientas de procesamiento que toman como entrada un archivo de video, y a través de ellas, se obtienen entidades reconocidas en distintos instantes de tiempo

dentro del mismo. Estas herramientas son:

- *Rekognition Video Features Label Detection*
- *Rekognition Video Features Celebrities Detection*
- *Clarifai Predict Videos*
- *Cloud Video Intelligence Label Detection*

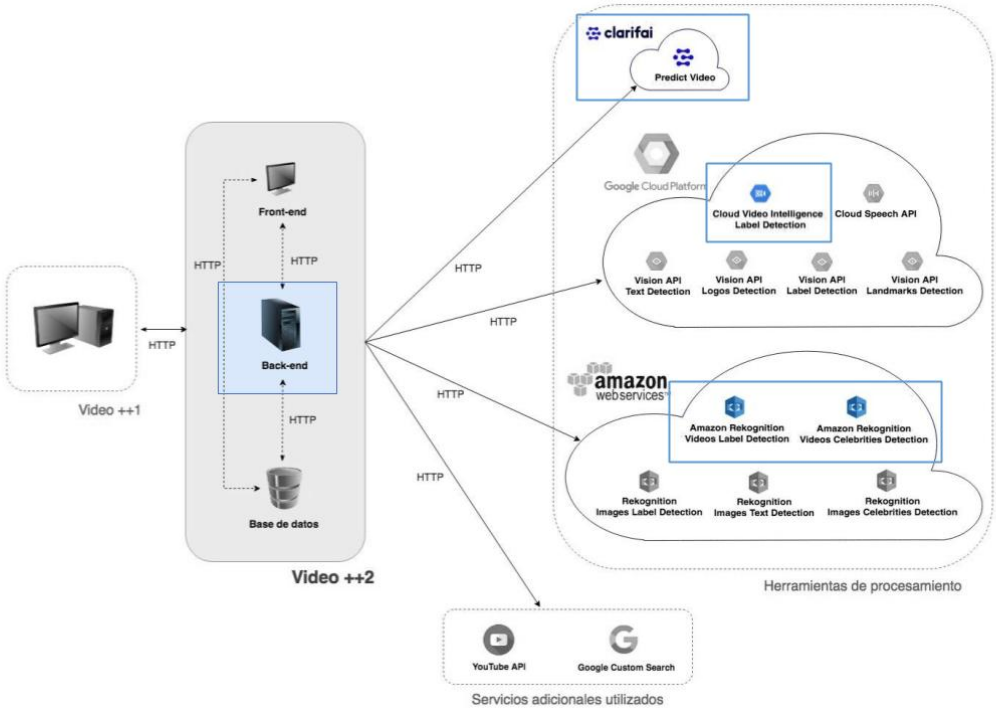


Figura 5.2: Arquitectura del sistema resaltando los componentes involucrados en el análisis de video

5.2.1.2. Análisis de audio

Con motivo de enriquecer la información obtenida para un video, se analiza también, el audio extraído del mismo. A partir de este se pueden detectar palabras clave provenientes por ejemplo, de diálogos, relatos, o canciones. La herramienta utilizada para el procesamiento de audio es la siguiente:

- *Cloud Speech API*

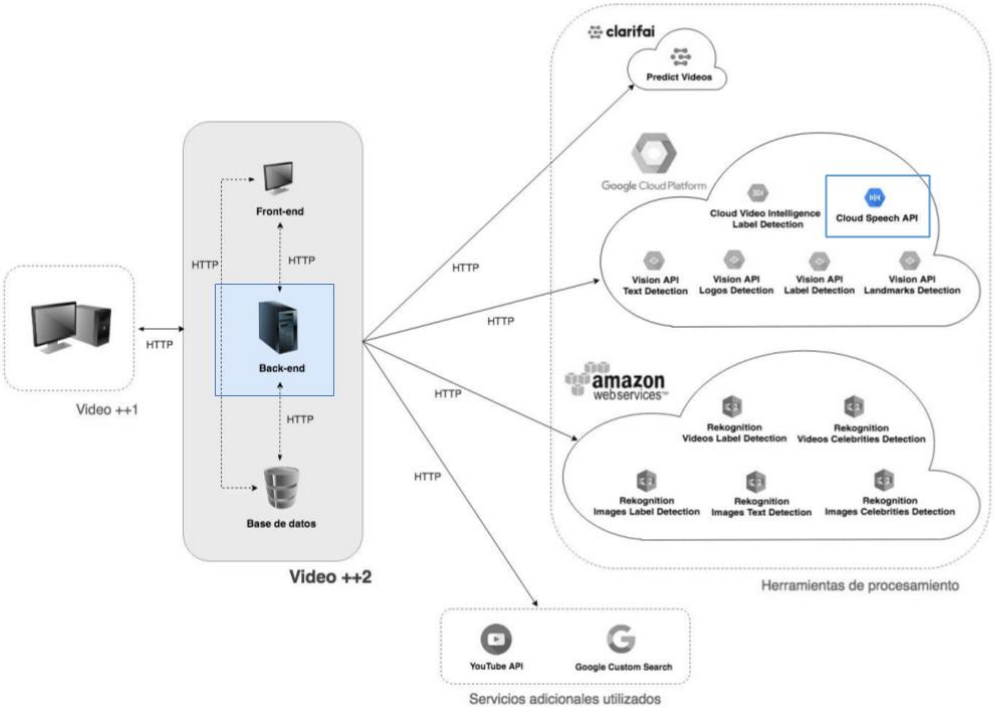


Figura 5.3: Arquitectura del sistema resaltando los componentes involucrados en el análisis de audio

5.2.1.3. Análisis de imágenes

Siguiendo con la premisa de obtener una mayor cantidad de información de interés sobre un video, se decidió capturar y analizar las imágenes correspondientes a ciertos momentos del mismo.

Determinar los instantes de tiempo en los cuales capturar las imágenes supone un desafío importante. Las imágenes elegidas repercuten directamente en la calidad de la información obtenida, ya que si estas no revelan contenido interesante, los datos generados a partir de ellas carecerían de valor.

Se adoptaron dos estrategias para resolver este problema:

- **Capturar imágenes según la duración del video:** los intervalos de tiempo en los que se extraerá una imagen, varían entre cinco segundos y un minuto, en donde los videos de menor duración son los que poseen la frecuencia más corta. Esta estrategia busca principalmente amortiguar costos (analizar menos imágenes implica menores costos) y prevenir tiempos de procesamiento sumamente extensos (procesar más imágenes aumenta los tiempos del análisis).

- **Capturar imágenes según los cambios de escena:** se busca extraer imágenes en los instantes donde se produce un cambio de escena en el video. Estos lapsos de tiempo son sumamente valiosos ya que, por lo general, el contenido visual detectado al comienzo de una escena no varía hasta que se pasa a la siguiente. Por lo tanto, con estas capturas se cubriría gran parte de la información proveniente de imágenes. Para encontrar estos momentos se utilizó la herramienta *Cloud Video Intelligence Shot Change Detection*, que

analiza el video y retorna los instantes de tiempo en donde se produce un cambio de escena.

Para lograr un balance entre cantidad y calidad de imágenes a procesar, se aplican las anteriores estrategias de manera combinada. Una vez conseguidas las capturas, son analizadas por las siguientes herramientas:

- *Rekognition Image Features Text Detection*
- *Rekognition Image Features Celebrities Detection*
- *Rekognition Image Features Label Detection*
- *Vision API Text Detection*
- *Vision API Label Detection*
- *Vision API Logos Detection*
- *Vision API Landmarks Detection*

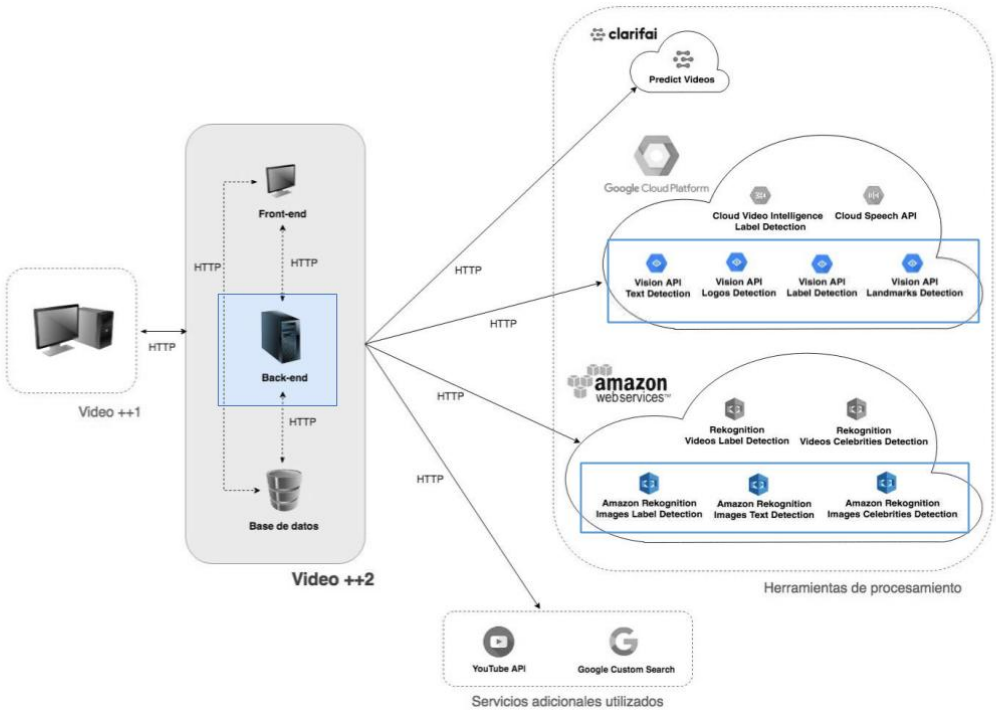


Figura 5.4: Arquitectura del sistema resaltando los componentes involucrados en el análisis de imágenes

5.2.2. Normalización de la información

Teniendo en cuenta la cantidad de herramientas externas con las que el sistema interactúa, y que los formatos de las respuestas de cada una varían ampliamente (ver formatos de respuestas en apéndice A), es importante normalizar la información obtenida y representarla en un

único formato. De lo contrario, la manipulación de los datos del sistema se volvería compleja y aumentaría la dificultad del mantenimiento de la información.

Por lo expresado en el párrafo anterior, luego de la generación de información, el sistema se encarga de adaptar la respuesta de cada herramienta a un formato común para todas. Este se corresponde con un listado de hitos, en donde cada hito contiene:

- elemento detectado
- instante de tiempo en donde ocurre
- duración
- nivel de confianza

El modelo de datos completo se detallará más adelante en el capítulo 6.

5.2.3. Filtrado de hitos

Una vez completada la generación y normalización de información vistas en las secciones anteriores, surge la necesidad de determinar qué hitos mantener y cuáles despreciar.

Este requerimiento nace por la intención de lograr un índice con información concreta y de la mejor calidad posible. Sin embargo, los resultados de las pruebas de concepto del capítulo 3 muestran que las herramientas utilizadas no están libres de errores. Ejemplos de esto pueden ser: entidades identificadas con bajo nivel de confianza (que pueden derivar en detecciones equivocadas), textos mal escritos o palabras que no aportan al contenido del índice.

Debido a esto, se plantea el diseño de un procedimiento cuyo objetivo sea el filtrado de este tipo de contenido en la medida que sea posible. Para lograr este cometido se definen dos tipos de filtros:

5.2.3.1. Filtro por nivel de confianza

Se define un umbral de aceptación para cada herramienta utilizada. Este indica el valor de confianza mínimo con el cual debe ser detectada una entidad para ser incluida en el índice generado. Esta definición se hizo en base a la evaluación de las respuestas de las herramientas vistas en el capítulo 3. Los niveles de confianza finalmente aceptados se pueden observar en el cuadro 5.1.

Este filtro se aplica a todas las herramientas a excepción de *Vision API Text Detection* que no proporciona niveles de confianza para sus detecciones.

Herramienta	Mínima confianza (%)
Imágenes	
Vision API - Label Detection	95
Vision API - Text Detection	85
Vision API - Logo Detection	90
Vision API - Landmark Detection	50
Rekognition Image Features - Label Detection	95
Rekognition Image Features - Text Detection	90
Rekognition Image Features - Celebrities Detection	90
Video	
Cloud Video Intelligence - Label Detection	88
Rekognition Video Features - Label Detection	90
Rekognition Video Features - Celebrities Detection	90
Predict Videos	99
Audio	
Cloud Speech API	80

Cuadro 5.1: Confianzas mínimas para las herramientas utilizadas

5.2.3.2. Filtro por texto irrelevante

Tiene como objetivo descartar palabras que no aportan valor al contenido, como artículos, pronombres, preposiciones, etc. Este tipo de palabras son llamadas "palabras vacías" o "stop words". [12]

Existen diversas listas de "stop words" para inglés y español disponibles en internet. Para generar este filtro, se utiliza una para cada uno de los dos idiomas. Toda palabra detectada que se encuentre dentro de alguna de ellas será eliminada. [13] [52]

5.2.3.3. Filtro por texto incorrecto

Mediante este procedimiento, se pretende descartar cadenas de caracteres sin sentido. Estos casos aparecen principalmente a través del uso de las funcionalidades de detección de texto, por lo que tiene sentido que el filtro sea utilizado únicamente sobre este tipo de herramientas.

En particular, se aplica sobre *Vision API Text Detection*, ya que para el resto de las herramientas de detección de texto se utiliza el filtro por nivel de confianza. A su vez, esta estrategia permite comparar cómo la calidad del índice generado cambia al aplicar diferentes filtros sobre los resultados de los diversos servicios utilizados.

A continuación se especifican las dos validaciones realizadas en este procedimiento de filtrado.

Determinar si el texto existe

Es posible que ocurran errores a la hora de detectar textos. Por ejemplo, las herramientas

pueden confundir o no reconocer ciertos caracteres, resultando en la generación de palabras o frases sin sentido. Estos casos, a su vez, se pueden dar cuando un texto al fondo de una toma es tapado por otros elementos, por lo que no se logra identificarlo en su totalidad. Para mitigar estos casos, se emplea el uso de un diccionario en donde se verifica que el texto detectado se corresponda con una palabra del idioma inglés o español [11].

Esto, sin embargo, no resuelve el problema en su totalidad. ¿Qué sucede con los nombres propios? ¿O con secuencias de caracteres que no se corresponden con una palabra pero igualmente tienen sentido? Un ejemplo de este caso podría darse en un tanteador de un partido de fútbol, donde se pueden detectar cadenas como "Pen 1 - 1 Nac". Si bien este texto no se encuentra en un diccionario, es una secuencia de caracteres que no se querría filtrar ya que en realidad tendría sentido e importancia a la hora de construir el índice. De aquí se desprende la segunda validación aplicada sobre los textos detectados.

Determinar si el texto tiene sentido

En el caso de que el texto no se encuentre en el diccionario, se realiza una validación extra. Esta pretende determinar si el texto en cuestión tiene sentido o relevancia, a pesar de no ser una palabra real.

Para ello, se utiliza la herramienta *Google Custom Search* [22] que permite crear un motor de búsqueda basado en las tecnologías ofrecidas por Google. De esta forma, se determina qué tanto significado tiene una secuencia de caracteres para un humano en base al número de resultados obtenidos con esta herramienta. Por ejemplo, para el caso del texto "Pen 1 - 1 Nac", una búsqueda en Google retorna más de 20 millones de resultados. [51] Esto da a entender que es un texto que realmente tiene sentido. Sin embargo, una secuencia de caracteres insignificante como "fsru2", retorna menos de 5000. [50] En base a esta observación se fija un umbral, y los textos cuyas búsquedas devuelvan una cantidad de resultados por debajo de ese límite son descartados.

Un beneficio extra que se obtiene a partir del uso de *Google Custom Search* es la conocida funcionalidad de Google "quizás quiso decir". Esta sugiere al usuario una búsqueda similar a la realizada aplicando correcciones sobre el texto inicial. De esta forma, sería posible corregir las etiquetas de los índices con alteraciones sufridas a la hora de detectar los textos. Por ejemplo, si el elemento detectado era "Barclna", esta funcionalidad devuelve los resultados asociados a la búsqueda "Barcelona", que son considerablemente mayores. Se asume entonces que, si existe un resultado similar propuesto por el buscador, se trata de que hubo alguna falla al detectar el texto y se aplica la corrección sugerida para el mismo.

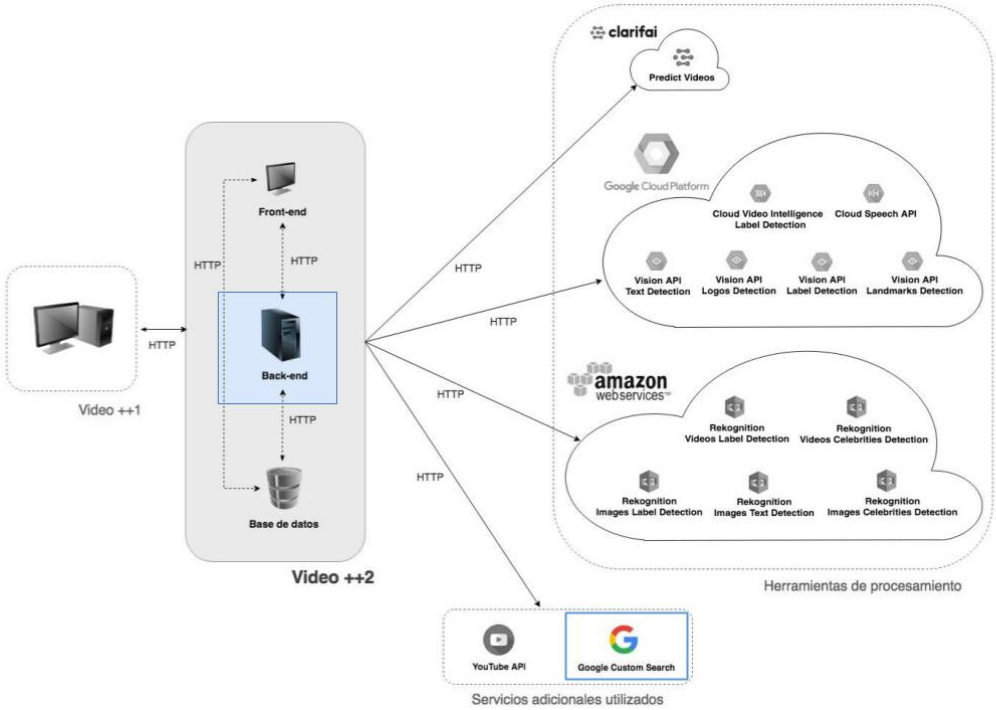


Figura 5.5: Arquitectura del sistema resaltando la comunicación con *Google Custom Search*

5.2.4. Unificación del índice

Una vez concluido el proceso de filtrado de hitos, se debe generar el índice final con la unión de los resultados provenientes de cada fuente de información. En este paso es necesario identificar y combinar los hitos que fueron detectados por más de una herramienta. Los detalles de cómo este mecanismo es llevado a cabo se especifican en el capítulo de Implementación.

5.2.5. Identificación de etiquetas relevantes por categoría

Se decidió desarrollar un módulo para la obtención y organización de información que pueda ser explotada en un marco de trabajo futuro. Una primera aproximación para ello es la construcción de una lista de entidades de interés por categoría de video. Esta idea se ve reflejada en el requerimiento funcional 4.2.2.1 descrito en el capítulo 4.

5.3. Front-end

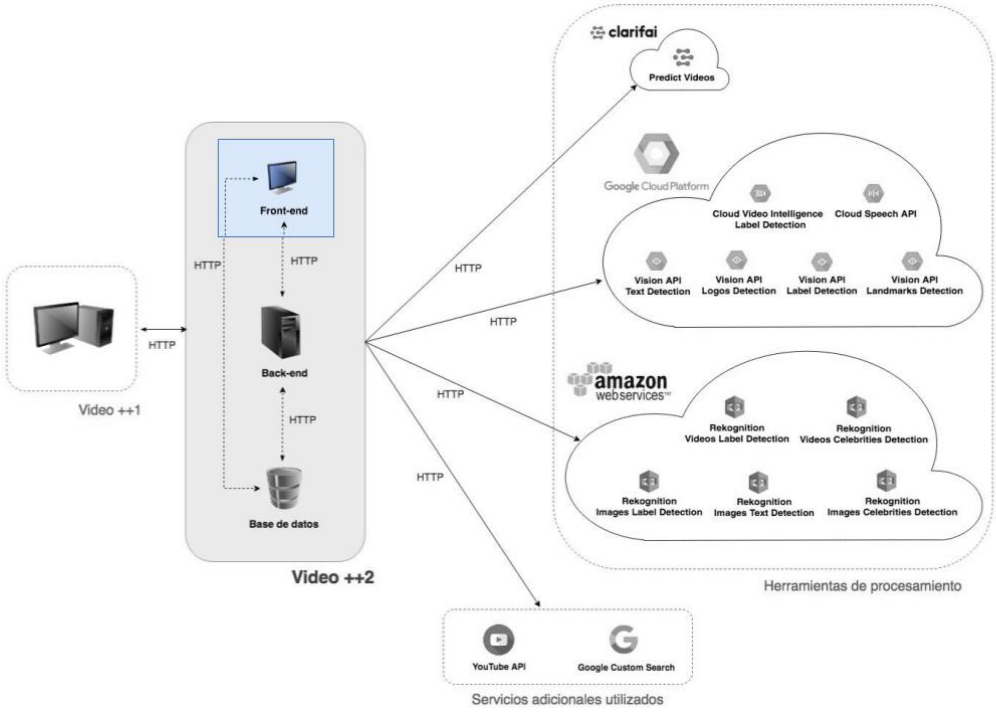


Figura 5.6: Arquitectura del sistema resaltando el front-end

Este módulo tiene como objetivo mostrar los resultados de los análisis realizados sobre los videos, así como también presentar los mismos de forma más amigable y organizada. Los requerimientos funcionales que debe cumplir el front-end fueron detallados en el capítulo 4, sección 4.2.1.4.

5.3.1. Vistas principales

- **Panel principal de un video:** esta vista despliega la información generada para un video. Aquí se observan datos como su título y categoría a la que pertenece, además de contar con el listado de las diferentes herramientas utilizadas para su análisis. Para cada una, se incluye el estado de procesamiento en el que se encuentra, además de poder acceder a los hitos que detectaron. Por último, en el caso de tratarse de un video ya procesado, se muestra el índice finalmente construido.

- **Historial de videos:** aquí se muestra un listado de los videos existentes en el sistema. Haciendo click sobre alguno de ellos, se lleva al usuario a la vista del panel para el video en cuestión.

5.4. Persistencia de datos

Para el almacenamiento de información, se decidió utilizar la base de datos no relacional Firebase. [15] El motivo de ello es que, en un principio, cumple los requisitos necesarios para soportar los volúmenes de información a manejar dentro del sistema. Asimismo, es una tecnología que ha ganado gran popularidad en la actualidad, lo que genera curiosidad sobre su desempeño y se desea aprovechar la oportunidad de evaluar sus beneficios en una actividad experimental.

Por otro lado, proporciona una serie de funcionalidades extra que resultan ser de utilidad para el prototipo a desarrollar. Una de ellas es la notificación instantánea a quien esté "suscrito" a novedades sobre los datos almacenados. Los "suscriptores" pueden ser tanto clientes web, aplicaciones móviles, o módulos de back-end. Estos son notificados al momento de detectar cambios en los datos, haciendo que no sea necesario refrescar una página web, o generar una nueva solicitud al servidor para mostrar la información actualizada. [17] Un ejemplo donde se puede apreciar el valor de esta funcionalidad es a la hora de mostrar los estados de procesamiento de cada herramienta de generación de índices, ya que en la interfaz de usuario se desplegará dicha información en tiempo real.

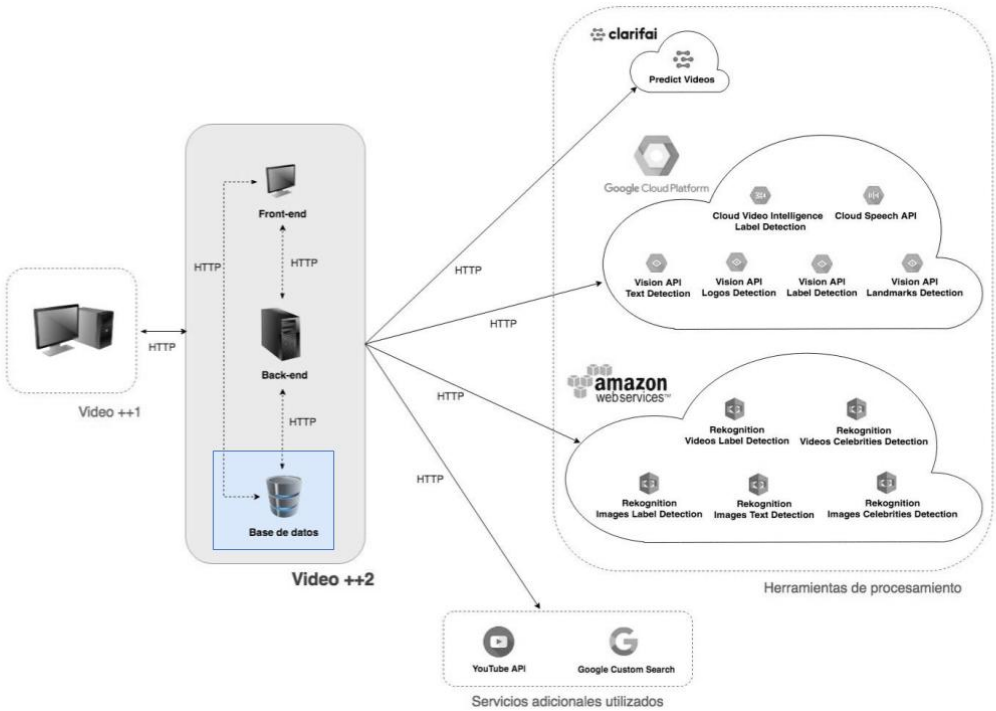


Figura 5.7: Arquitectura del sistema resaltando la base de datos

Capítulo 6

Modelo de datos

Como se adelantó en el capítulo anterior, dada la diversidad de servicios con los que este proyecto interactúa, se desprende la necesidad de definir un modelo de datos que se ajuste de la mejor manera a los formatos de respuesta de cada uno. A su vez, es importante que el modelo exprese la información de forma comprensible y sin ambigüedades.

Para ello, se definió un conjunto de entidades que serán detalladas en las siguientes secciones. El contenido de cada una de ellas se encuentra almacenado en Firebase y se estructuran como objetos en formato JSON.

6.1. Representación de las entidades

6.1.1. Información del video

Esta entidad representa a un video dentro del sistema. La información de un video se identifica de forma unívoca mediante una clave generada a partir de su URL. Se modela de la siguiente manera:

- **title**: título del video.
- **category**: almacena el identificador de la categoría a la que el video pertenece.
- **source**: indica el origen del video. Este campo puede adoptar dos valores; "youtube" para videos provenientes de esta fuente o "Public video" para el resto.
- **date_created**: fecha de creación del análisis del video.
- **last_updated**: fecha de última modificación del análisis del video.
- **status**: estado en el que se encuentra el análisis del video. Los posibles valores para este campo se encuentran definidos en la sección 4.2.1.3 del capítulo 4.
- **video_id**: identificador del video en YouTube. Este campo aplica únicamente para videos provenientes de dicha fuente.

- **shot_changes**: almacena los instantes (en milisegundos) en los que se detecta un cambio de escena en el video.

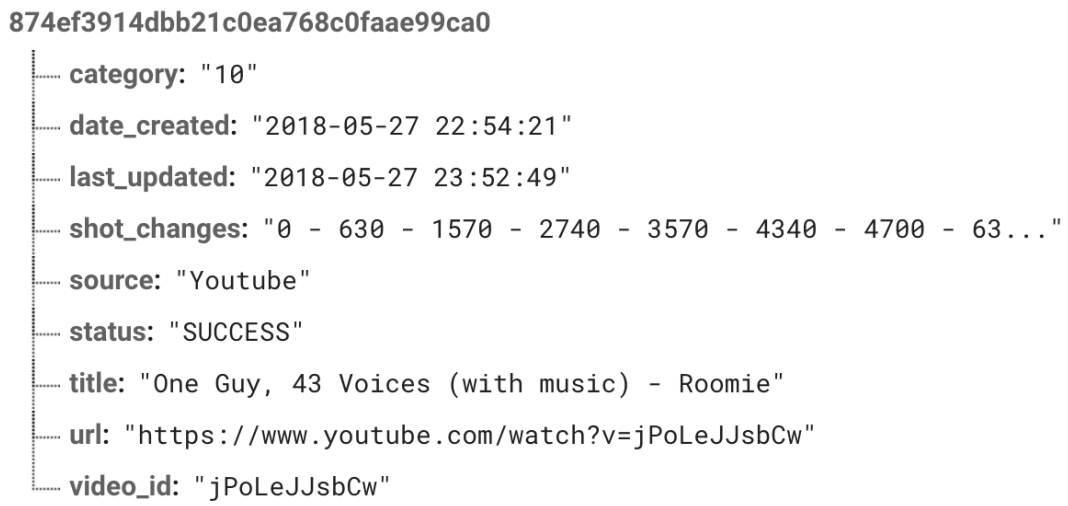


Figura 6.1: Representación en Firebase de la información del video: *One Guy, 43 Voices* [58]

6.1.2. Análisis del video

Una de las necesidades principales del proyecto, es la de reunir la información acerca del análisis de un video. Esta entidad, es la que representa los resultados obtenidos por cada servicio utilizado y el índice finalmente generado.

El análisis de un video consta de dos elementos fundamentales:

- Resultados obtenido por cada servicio
- Resumen de los resultados obtenidos

Se pasará a explicar en detalle ambos puntos.

6.1.2.1. Resultado obtenido por servicio

Aquí se modela la forma estándar de almacenar los resultados obtenidos por cada uno de los servicios de procesamiento. Estos se diferencian a través de un identificador único y organizan la información de la siguiente manera:

- **data:** lista de etiquetas detectadas por el servicio en cuestión. Cada elemento de esta lista tiene asociado los instantes de tiempo (en milisegundos) en los cuales aparece dicha etiqueta, además de la confianza con la cual fue detectada.
- **information:** contiene el estado en el que se encuentra el análisis para este servicio, junto con su fecha de creación y fecha de última actualización.

En la figura 6.2 se observa el modelo representado en Firebase. La raíz de la estructura se corresponde con el identificador del servicio y por debajo se desprende el resto de la información.

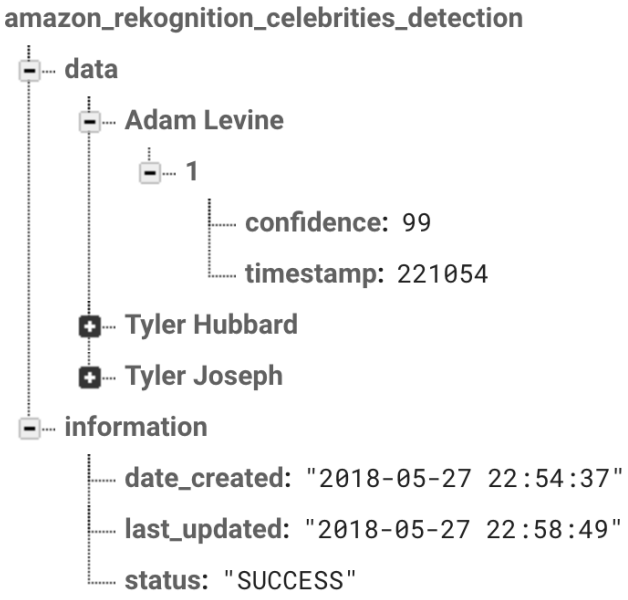


Figura 6.2: Representación en Firebase de los resultados obtenidos para el cliente *Rekognition Image Features Celebrities Detection*

6.1.2.2. Resumen de los resultados obtenidos

Aquí es donde se unifican los resultados obtenidos en cada servicio. Esta entidad es la que contiene el índice finalmente generado que será consumido por el usuario. La estructura se compone de los siguientes elementos:

- **data:** Contiene una lista con todos los hitos generados en base a los datos provistos por los diferentes servicios. Cada elemento de la lista, agrupa la información de la etiqueta detectada, junto con el instante en dónde aparece (start) y su duración (duration).
- **information:** Esta entidad se corresponde con la información del procesamiento del video a nivel general. Aquí se encuentra el estado en el que se encuentra el análisis y datos de creación y última actualización del mismo.

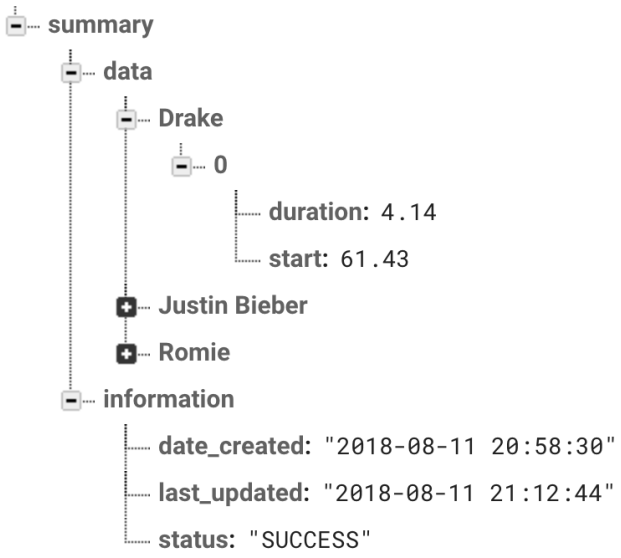


Figura 6.3: Representación en Firebase de un extracto del resumen para el video *One Guy, 43 voices* [58]

6.1.3. Información del servicio

Esta entidad surge con el objetivo de mantener información sobre los servicios de procesamiento de forma independiente a otras entidades. Si bien dentro del alcance del proyecto interesa almacenar únicamente la confianza mínima aceptable de cada uno, se optó por mantener esta información desacoplada, previendo que en un futuro, se puedan agregar datos de manera sencilla y sin necesidad de modificar el diseño original.

Por otro lado, este modelo brinda una mayor flexibilidad de configuración ya que da la posibilidad de modificar los umbrales de confianza a demanda, sin necesidad de generar versiones nuevas de la aplicación ante un cambio de este tipo.

Cada servicio se diferencia a través de un identificador único y contiene la siguiente información:

- **minimum_confidence:** valor que determina la confianza mínima que debe tener un elemento detectado por el servicio para ser incluido en el índice generado (ver sección 5.2.3.1).

Figura 6.4: Representación en Firebase de la información del servicio *Rekognition Image Feature Label Detection*

6.1.4. Categoría

Esta entidad representa una categoría de video. Su importancia radica en que en ella se incluye la lista de etiquetas interesantes para los usuarios (requerimiento funcional 4.2.2.1). Cada categoría se identifica a través de un identificador numérico. Para cada una, se almacenan los datos a continuación:

- **name:** nombre de la categoría.
- **white_list:** lista de etiquetas relevantes para la categoría. Cada etiqueta relevante se representa de la siguiente manera:
  - **tag:** palabra relevante.
  - **times_liked:** cantidad de veces que la palabra fue marcada por un usuario como relevante.

Figura 6.5: Representación en Firebase de la categoría “entretenimiento”

6.1.5. Resultados de búsqueda de Google Custom Search

El propósito de esta entidad, es almacenar los resultados obtenidos por la API de *Google Custom Search* para una búsqueda. El objetivo que persigue esto, es minimizar el volumen de solicitudes que se realizan hacia dicho servicio. De esta forma, antes de generar un pedido a la API, se consulta en Firebase si la búsqueda en cuestión ya se encuentra almacenada.

Para ello, cada búsqueda se identifica mediante la(s) palabra(s) que la compone(n). El modelo se define de la siguiente manera:

- **result_count:** cantidad de resultados obtenidos para la búsqueda.
- **results_for:** palabras por las cuales se realizó la búsqueda. Esto se debe a que la API aplica correcciones sobre las palabras y efectúa la búsqueda sobre dichas correcciones.

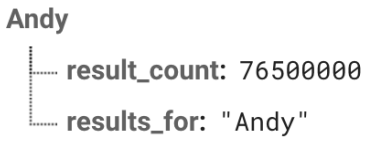


Figura 6.6: Representación en Firebase del resultado de la búsqueda “Andy” en Google Custom Search

Capítulo 7

Implementación

En el presente capítulo se profundiza sobre la implementación de la solución descrita previamente. El mismo se encuentra organizado en secciones que se corresponden a los módulos principales del sistema. En cada una de estas se explica cómo fueron resueltas las distintas funcionalidades del sistema y, en algunos casos, decisiones de implementación orientadas a satisfacer requerimientos no funcionales.

7.1. Back-end

En esta sección, se exponen las tecnologías utilizadas para la construcción del módulo de back-end. Además, se describe cada uno de sus componentes, seguido por la resolución a nivel técnico de las principales funcionalidades. Finalmente, se trata la interacción con el proyecto Video++1.

7.1.1. Tecnologías utilizadas

El back-end fue desarrollado usando el lenguaje de programación Python en su versión 3.6.4 y se encuentra disponible a través del servicio Google App Engine.

7.1.1.1. Python

Se seleccionó este lenguaje para el desarrollo del back-end por diversas razones. Una de las más importantes, fue que los miembros del grupo ya contaban con experiencia sobre su uso. En este punto, se consideró que no aportaría tantos beneficios atravesar una curva de aprendizaje más pronunciada al incursionar en nuevos lenguajes.

Otros dos puntos que inclinaron la balanza a favor de Python, fueron la legibilidad del código y la cantidad de bibliotecas compatibles disponibles. Este último punto es realmente importante, ya que el uso de bibliotecas facilita la resolución de diversas acciones de manera sencilla. Algunas de estas acciones son, por ejemplo:

- Descarga de videos a través de una URL.

- Extracción de imágenes de un video.
- Clientes para conexión con los servicios de procesamiento.
- Comunicación con API de YouTube y Firebase.

Adicionalmente, existe una gran variedad de bibliotecas relacionadas al área de aprendizaje automático, dejando la puerta abierta a la adición de nuevas funcionalidades que exploten los beneficios de este paradigma.

Por otro lado, como se verá más adelante, fue necesario manejar distintos hilos de ejecución para poder ejecutar el procesamiento de datos en paralelo y, de esta forma, obtener resultados en tiempos más cortos. En este rubro, Python provee un manejo de hilos que abstrae al programador del uso de estructuras complejas para su implementación.

Otros beneficios que merecen ser resaltados son, la simplicidad con la que se puede levantar la aplicación en un ambiente local (ver manual del programador en apéndice D) y la amplia comunidad de desarrolladores que permite evacuar dudas a través de foros de manera ágil.

7.1.1.2. Google App Engine

Google App Engine es el servicio de alojamiento web donde se encuentra disponible el módulo de back-end. Uno de los principales motivos por el cual se eligió, es la posibilidad de desplegar fácilmente un aplicativo escrito en el lenguaje Python. Otra razón para la selección de esta tecnología fue el crédito otorgado por Google, el cual puede ser utilizado tanto para las herramientas de procesamiento como para la infraestructura requerida.

Cabe mencionar que, además de Google App Engine, se evaluó el servicio de infraestructura Google Compute Engine. La diferencia entre ambos, es que el segundo provee máquinas virtuales, donde el desarrollador puede configurar el entorno que desee. En contrapartida, Google App Engine es un servicio de alojamiento, donde solamente es necesario subir la versión de la aplicación que se desea liberar y esta resuelve automáticamente las configuraciones relacionadas a las necesidades de cómputo. Esto es, por ejemplo, aumentar el número de máquinas disponibles y balancear la carga entre ellas para mejorar el rendimiento del sistema.

Las características anteriores influyeron en la decisión tomada de la siguiente manera. Ocurre que algunas de las funcionalidades del sistema, como el cambio de formato de archivos (necesario para el procesamiento de audio), requieren de la instalación de otros componentes. Esta actividad es más sencilla cuando se cuenta con una máquina virtual con un sistema operativo instalado y el programador se encarga de agregar las dependencias y configuraciones necesarias para su funcionamiento (como sucede en Google Compute Engine). Sin embargo, esto genera que se inviertan varias horas en tareas de configuración de entorno, además de perder las funcionalidades de escalado automático de Google App Engine.

Afortunadamente, existe una versión de Google App Engine que junta "lo mejor de los dos mundos". Este es el ambiente flexible de Google App Engine, el cual mantiene los beneficios de escalabilidad automática y facilidad de manejo de versiones, pero además admite

personalizar el ambiente donde se ejecuta la aplicación. Luego de varias pruebas, se lograron realizar las configuraciones necesarias, permitiendo satisfacer las necesidades relacionadas con las funcionalidades del sistema y la capacidad de cómputo e infraestructura.

7.1.2. Componentes de back-end

Con el objetivo de desarrollar una aplicación mantenible y modular, se optó por una implementación en capas. Se define una capa de controladores, una de servicios y una de manejadores de herramientas. Adicionalmente, se cuenta con un módulo en el que se encapsula la lógica y configuración para la manipulación y persistencia de datos. También, se desarrolla un conjunto de utilitarios que se encargan de resolver cuestiones como la descarga de videos, conversión de formatos de archivos y extracción de imágenes y audio de un video. Esta organización del código busca reducir la dificultad de agregar nuevas funcionalidades y fuentes de datos para la generación de índices.

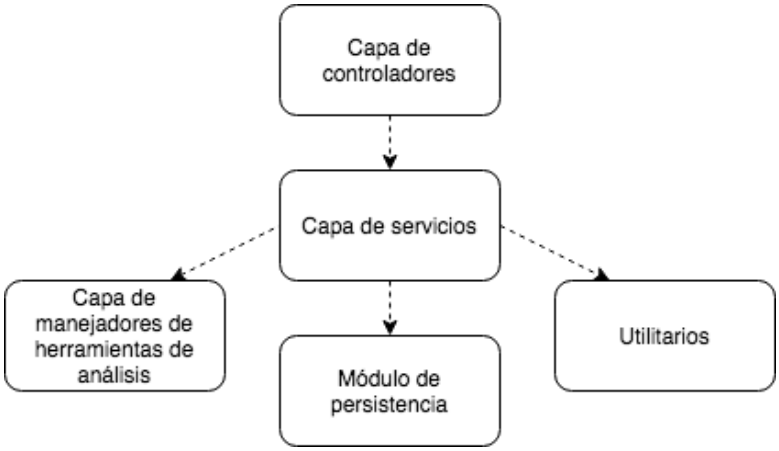


Figura 7.1: Diagrama de componentes de back-end

7.1.2.1. Capa de controladores

Aquí se encuentran los métodos a través de los cuales se disponibiliza la API del sistema. Esta capa define las URLs y parámetros necesarios para que otras aplicaciones puedan consumir las funcionalidades ofrecidas. Cuenta con dos controladores:

- *Controlador principal (Main Controller)*

  Encargado de ofrecer las funcionalidades de generación y obtención de índices para los videos.

- *Controlador de categorías (Categories Controller)*

  Responsable de publicar las operaciones de agregado y modificación de categorías en el sistema. Asimismo, expone la funcionalidad de dar "me gusta" a hitos en los videos, repercutiendo en el listado de hitos más relevantes por categoría.

7.1.2.2. Capa de servicios

Esta capa contiene la lógica de negocio de la aplicación. Se encarga de procesar información para completar las solicitudes recibidas por la capa de controladores. Para lograrlo, hace uso de las funcionalidades provistas por la capa de manejadores de herramientas y de los utilitarios.

Cada servicio posee una responsabilidad específica. Se detallan los mismos a continuación:

- *Servicio principal (Main Service)*

  Encargado de orquestrar el resto de los servicios de la capa.

- *Servicio de categorías (Categories Service)*

  Gestiona las categorías de videos existentes. Además, se ocupa de actualizar las listas de hitos más relevantes por categoría.

- *Servicio de video (Video Service)*

  Se encarga del procesamiento de los videos para poder obtener la información necesaria para la generación de los índices. Estos datos son obtenidos a partir de la interacción con los proveedores de servicios de análisis de video, encapsulados dentro de la capa de manejadores de herramientas.

- *Servicio de imagen (Image Service)*

  Se ocupa del procesamiento de las imágenes extraídas de un video para recabar información a utilizar para la creación de los índices. Hace uso de los manejadores de herramientas para establecer la comunicación con los proveedores de servicios de análisis de imágenes.

- *Servicio de audio (Audio Service)*

  Responsable del procesamiento del audio extraído de un video con el fin de utilizar los resultados en la generación de los índices. Al igual que en los servicios de imágenes y videos, se utiliza la capa de manejadores de herramientas para realizar las solicitudes a los servicios que extraen el texto de un audio.

- *Servicio de filtrado (Filter Service)*

  Su objetivo es el filtrado de la información obtenida por los servicios anteriores para eliminar detecciones incorrectas e hitos irrelevantes.

- *Servicio de búsquedas (Search Service)*

  Se encarga de la comunicación con la herramienta *Google Custom Search* para obtener los resultados de búsqueda sobre los textos detectados.

7.1.2.3. Capa de manejadores de herramientas

Esta capa se ocupa de obtener la información requerida por la capa de servicios. En primer lugar, se comunica con las diversas herramientas de procesamiento para obtener los datos del análisis de video, audio e imágenes. Luego, se encarga de normalizar los resultados obtenidos para que se ajusten al modelo de datos definido en el capítulo 6.

Se define un manejador por proveedor existente (hasta el momento Google, Amazon y Clarifai) y por tipo de procesamiento (audio, imagen, video). Entonces, por ejemplo, para procesar imágenes utilizando las funcionalidades ofrecidas por Google, se tiene el *manejador de servicios de imágenes de Google*. Lo mismo aplica para el resto.

De aquí se desprende que, para agregar un nuevo proveedor al sistema, lo único que se necesita es crear un manejador para el tipo de procesamiento que se quiere realizar (implementando dentro la lógica necesaria) y que este sea consumido por alguno de los servicios descritos en la sección anterior.

7.1.2.4. Módulo de persistencia

Este módulo se creó con el objetivo de centralizar la configuración y lógica de comunicación con la base de datos (Firebase). Proporciona métodos mediante los cuales se pueden consultar y modificar las entidades del modelo de datos de forma simple. De este modo, cada operación que requiera de modificaciones sobre los datos, utiliza este módulo.

7.1.2.5. Utilitarios

Se crearon una serie de utilitarios que ayudan en aspectos específicos de la implementación. A continuación se detalla cada uno de ellos:

- *Operaciones sobre videos (Video Helper)*

  Provee las operaciones de descarga y borrado de un video en el sistema de archivos local. Para la descarga se utilizan las librerías *pytube* [47], para videos provenientes de YouTube, y *urllib* [54], para videos disponibles en una URL pública.

- *Operaciones sobre imágenes (Image Helper)*

  Expone las operaciones de extraer las imágenes de un video, tanto por frecuencia de tiempo, como por cambios de escena. La toma de capturas se realiza a través de la librería *opencv-python* [38].

  Además, en este utilitario se ofrece una función que elimina las imágenes del sistema de archivos local. La misma es invocada al final del análisis de un video.

- *Operaciones sobre audio (Audio Helper)*

  Este utilitario provee las operaciones necesarias para obtener el audio de un video. Cabe destacar que este debe ser extraído en el formato *.flac*, ya que así lo requiere la herramienta *Cloud Speech API* para procesar el audio.

Las dos tareas anteriores fueron resueltas a través del programa *Ffmpeg* [14], que permite la extracción del audio en el formato deseado. *Ffmpeg* no es parte de las librerías para Python, sino que es un software en sí mismo que debe ser ejecutado simulando su invocación. Esto se hace utilizando la librería de Python, *os*, que permite realizar operaciones como si se ejecutaran en la línea de comandos de la terminal. En la invocación de *Ffmpeg* es necesario pasar como parámetros, la ruta del archivo de video fuente y la del archivo de audio destino con la extensión deseada.

Otra funcionalidad implementada en este utilitario es el borrado del audio del sistema de archivos local, invocada al final del análisis del mismo.

- *Conversor* XML *(XML encoder)*

  Debido a que Video++1 requiere que los datos de los índices les sean enviados en formato XML, surge la necesidad de convertir la información (generada como JSON) a dicho formato. Este utilitario permite realizar la conversión a través del uso de la librería *xml* de Python. [62]

7.1.3. Funcionalidades implementadas

7.1.3.1. Proceso de generación de índices

En la figura 7.2, se presenta un esquema en donde se desglosan las principales tareas llevadas a cabo en el proceso de generación de índices.

En primer lugar, se inicializa el análisis del video. Luego, se ejecutan las acciones de generación y filtrado de información en forma paralela para el video, su audio e imágenes extraídas. Una vez que las tres tareas son completadas, se unifica la información obtenida, y, por último, se finaliza el análisis.

A continuación, se pasa a describir en detalle cada uno de los pasos involucrados.

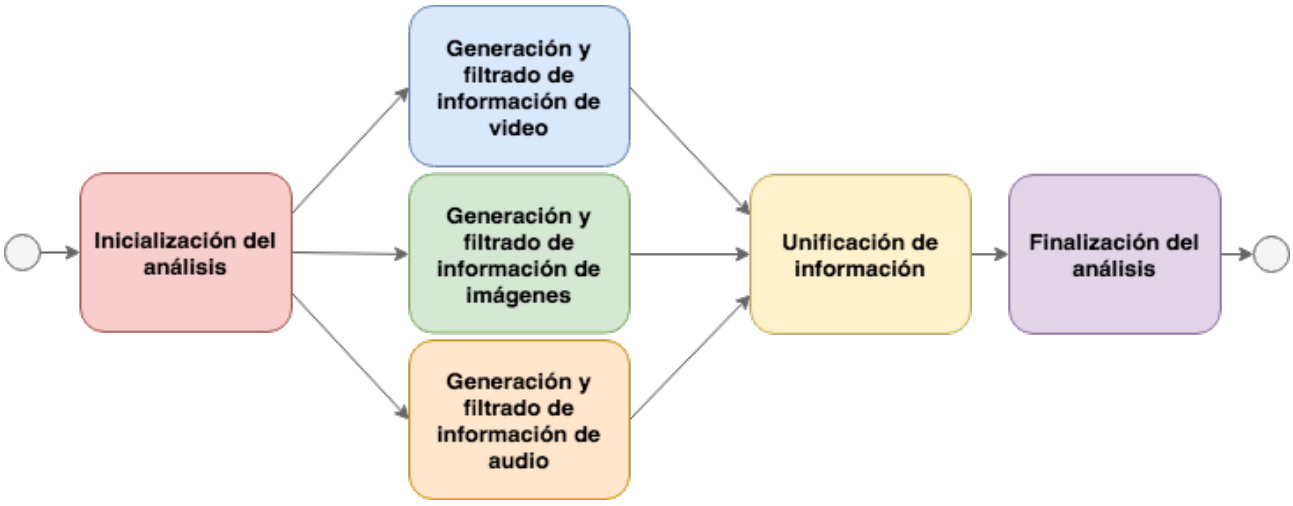


Figura 7.2: Proceso de generación de índices

Es importante destacar que las bifurcaciones en los diagramas del presente capítulo, indican que las tareas son ejecutadas en paralelo.

Inicialización del análisis

En primer lugar, para dar inicio al análisis, se debe cumplir que no se cuente con un índice ya creado en el sistema para ese video. En tal caso, se retorna el índice almacenado, a modo de no procesar información de forma innecesaria.

El proceso de generación de índices comienza cuando llega una petición de análisis de un video al Controlador Principal. Se debe pasar, de forma obligatoria, la URL del video a procesar en el parámetro `video_url`. Opcionalmente, se puede especificar el lenguaje del video a través del parámetro `language`, siendo tomado por defecto el inglés.

Una vez obtenidos los parámetros, se crea un identificador único para el video en base a su URL. Para esto, se utiliza el algoritmo MD5 utilizando el parámetro `video_url`. Seguidamente, se descarga el video, y se almacena temporalmente en un directorio interno del proyecto.

En caso de que se trate de un video proveniente de YouTube, se obtienen datos extra como su título, duración y categoría a la que pertenece. Esto se hace a través de una API provista por YouTube. [63]

Finalmente, se prosigue con el procesamiento del video, de las imágenes y del audio. Estas tres acciones son ejecutadas en diferentes hilos de forma paralela. La capa responsable de llevar a cabo dichas tareas es la de servicios.

Generación y filtrado de información de video

A continuación, se muestran los pasos a seguir para obtener los resultados de los análisis de las herramientas de procesamiento de videos:

1. Subida de video a repositorios en la nube

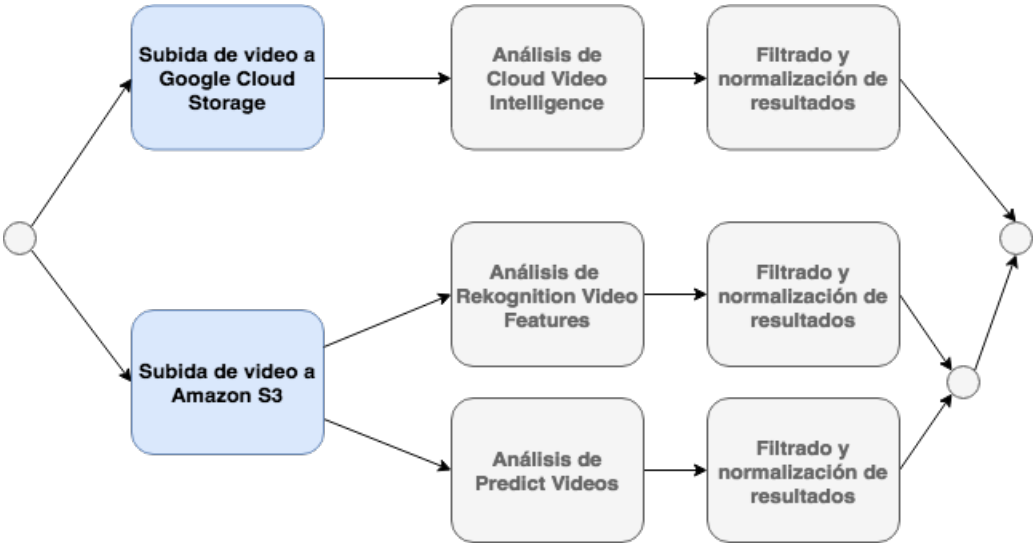


Figura 7.3: Generación y filtrado de información de video: subida a repositorios

Las herramientas de procesamiento de video utilizadas requieren que este se encuentre almacenado en un repositorio en la nube. Por ejemplo, para el caso de *Cloud Video Intelligence* se debe subir el video a Google Cloud Storage, mientras que *Amazon Rekognition Video Features* y *Clarifai Predict Videos* lo consumen de Amazon S3.

Para llevar esto a cabo, en primer lugar, el sistema debe autenticarse contra los respectivos repositorios. Esto se hace utilizando una clave secreta otorgada por el proveedor de servicio de almacenamiento. Luego, para subir el video, se utilizan las librerías *boto3*[1] y *google-cloud*[2].

Un punto que vale la pena mencionar, es que para acceder al video subido a Google Cloud Storage se requiere una autenticación previa, mientras que para Amazon S3 el video queda accesible a través de una URL pública. Esto se debe a que el único consumidor de la información almacenada en el repositorio de Google es *Cloud Video Intelligence*. Sin embargo, el video subido a Amazon S3 es utilizado por *Rekognition Video Features* y por *Predict Videos* por lo que es necesario que los archivos se encuentren disponibles públicamente.

[1]Librería de Amazon Web Services para Python que permite, a través de una API, hacer uso de servicios como Amazon S3. [6]

[2]Cliente para Python que permite hacer uso de los servicios ofrecidos por Google Cloud Platform a través de una API. [19]

2. Interacción con herramientas de análisis

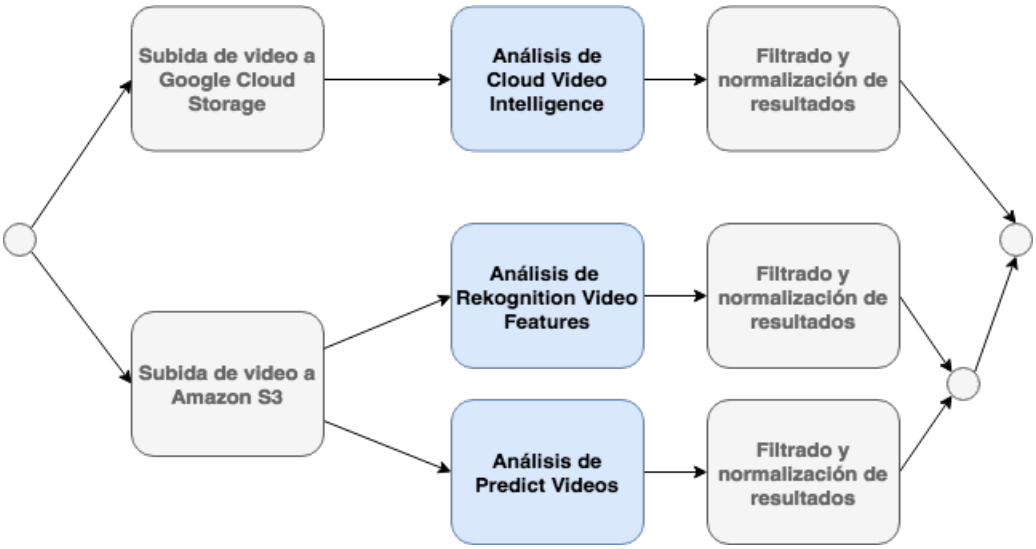


Figura 7.4: Generación y filtrado de información de video: interacción con las herramientas de análisis

En este paso se establece la comunicación con las herramientas de procesamiento de videos para obtener los resultados del análisis.

Previo a esto, es necesario autenticarse contra cada una. Para ello, en el caso de Amazon y Google, se utilizan las ya mencionadas librerías *boto3* y *google-cloud* respectivamente. Por otro lado, la autenticación con Clarifai se hace a través de la librería *clarifai*. [8]

3. Filtrado y normalización de resultados

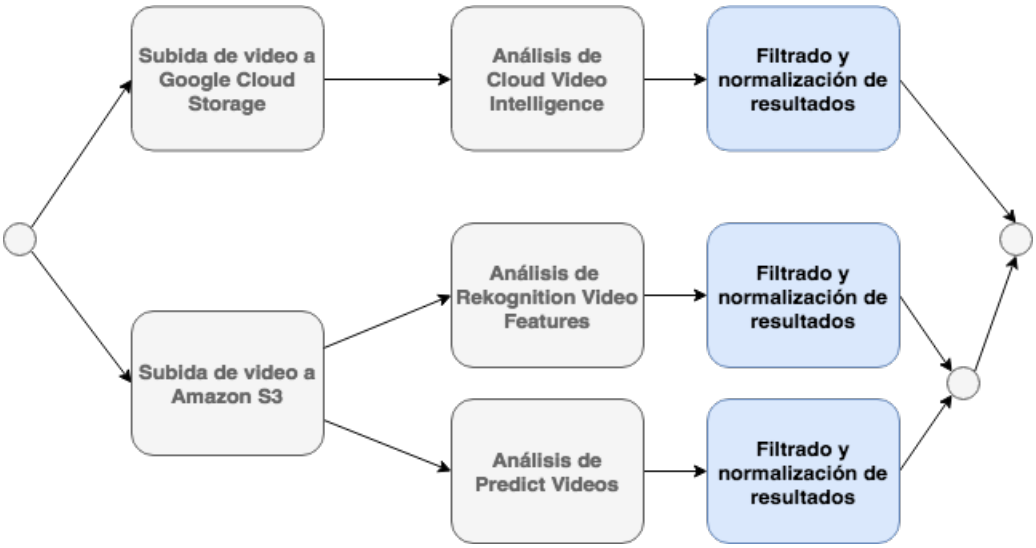


Figura 7.5: Generación y filtrado de información de video: filtrado y normalización de resultados

En esta etapa del proceso, se realizan dos tareas fundamentales. Una de ellas, es la normalización de cada respuesta obtenida al formato estándar definido en el capítulo 6, sección

6.1.2.1. Cada herramienta cuenta con su respectiva implementación para normalizar los resultados, ya que cada una retorna los datos en un formato diferente.

La otra, se refiere al filtrado de la información que no cumpla con un cierto nivel de calidad. Para esto, se implementa el filtro por confianza definido en la sección 5.2.3.1 del capítulo 5.

La información de la confianza mínima para cada herramienta es almacenada en Firebase. A la hora de aplicar este filtro, se obtiene de la base de datos dicho valor y se eliminan los elementos detectados cuya confianza se encuentre por debajo de este.

Generación y filtrado de información de imágenes

Aquí se especifican las tareas seguidas para completar el proceso de generación y filtrado de información a partir de las imágenes capturadas para el video.

1. **Generación de imágenes a procesar**

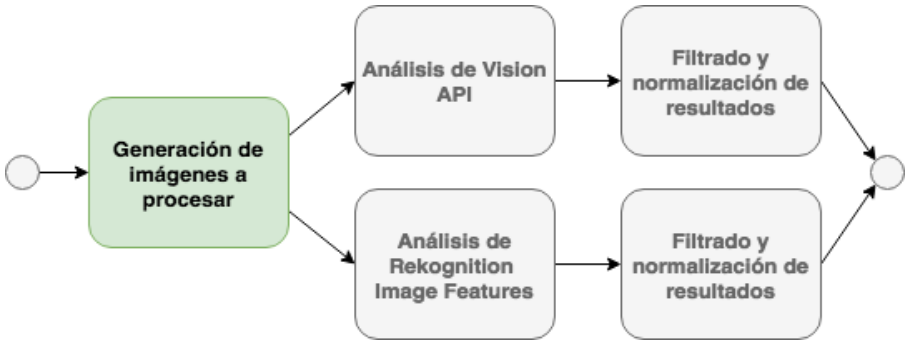


Figura 7.6: Proceso de generación de índices a partir de imágenes del video: generar imágenes a procesar

El primer paso dentro de este flujo, trata de determinar las capturas del video que se pretenden analizar. Para esto, se definieron dos criterios ya especificados en el capítulo 5, sección 5.2.1.3. A continuación sus detalles de implementación:

- **Capturas de imágenes según período de tiempo**

  Se definió un criterio mediante el cual se define el intervalo de tiempo que separa una captura de imagen de otra, en base a la duración del video. El mismo se representa a continuación en forma de tabla:

Duración del video (minutos)	Intervalo de tiempo para las capturas (segundos)
0 - 5	5
5 - 10	10
10 - 30	60

Cuadro 7.1: Intervalos de capturas de imágenes según duración del video

- **Capturas de imágenes según cambios de escena en el video**

  Para la implementación de este criterio, se utiliza la herramienta *Cloud Video Intelligence Shot Change Detection*. La comunicación con la misma se realiza a través de la librería *google-cloud*, mencionada anteriormente.

  Una vez obtenidos los resultados, se tiene una lista con los instantes de tiempo en donde se detectó un cambio de escena en el video en cuestión.

Luego de determinar los momentos del video para la captura de imágenes, se hace uso de las funcionalidades implementadas en el utilitario, ImageHelper, para extraer dichas capturas del video. Una vez generadas estas imágenes, son almacenadas en un directorio local al proyecto para que puedan ser accedidas por las diferentes herramientas de análisis.

2. Interacción con las herramientas de análisis

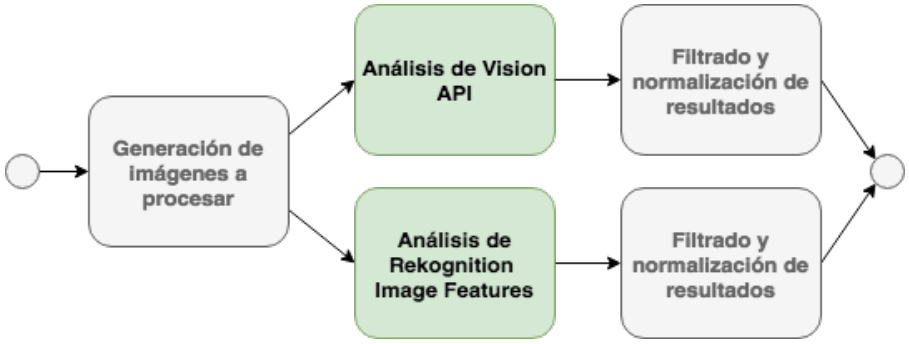


Figura 7.7: Proceso de generación de índices a partir de imágenes del video: interacción con las herramientas de análisis

La interacción con las herramientas de análisis de imágenes se realiza a través de las librerías *google-cloud* y *boto3* en el caso de Google y Amazon respectivamente. Como ya se mencionó previamente, se encargan, en primer lugar, de resolver la autenticación. Luego, de realizar las solicitudes de procesamiento y obtener las respuestas requeridas.

3. Filtrado y normalización de resultados

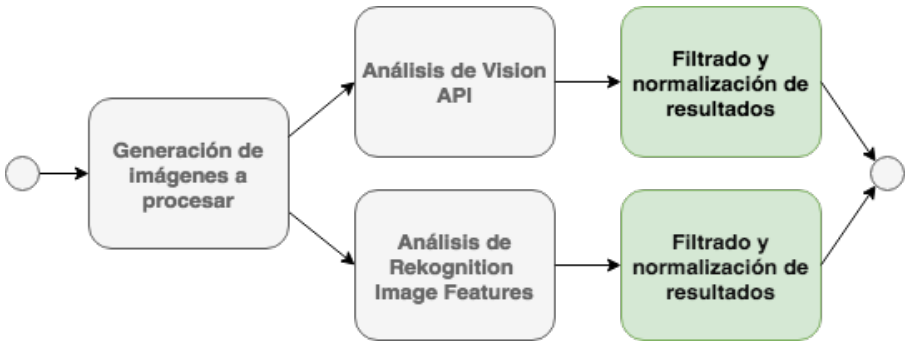


Figura 7.8: Proceso de generación de índices a partir de imágenes del video: filtrado y normalización de resultados

Como se adelantó en el capítulo de Diseño de la solución, en el procesamiento de imágenes es donde se implementan los algoritmos más elaborados de filtrado. Esto se debe a que es el paso en donde se introducen la mayoría de las detecciones erróneas. Estas surgen, más que nada, a través de los servicios de reconocimiento de texto.

En esta sección, se profundiza sobre la implementación de las distintas estrategias de filtrado aplicadas a los textos detectados por las herramientas correspondientes.

- **Filtrado por confianza**

  Su funcionamiento ya fue detallado en secciones anteriores. Este filtro es aplicado a todas las herramientas de análisis de imágenes, a excepción de *Vision API Text Detection*, que no retorna una confianza para sus detecciones.

- **Filtrado de texto incorrecto**

  Se refiere al filtro por palabras existentes en un diccionario. Para implementarlo, las palabras del diccionario son almacenadas en Firebase. Cada vez que se requiera saber si un texto detectado se corresponde con una palabra, se busca dicho texto en la base. En el caso de encontrarlo, se lo toma como válido. De lo contrario, se pasa a aplicar el siguiente filtro.

- **Filtrado de texto sin sentido**

  Este filtro se aplica luego del anterior, para no descartar texto que no esté dentro de un diccionario pero que puede tener sentido y ser importante.

  Como ya fue introducido en el capítulo de Diseño de la solución, la manera de implementar este filtrado es a través de la herramienta *Google Custom Search*. Su funcionamiento se base en utilizar el motor de búsqueda para determinar si el texto detectado retorna una cantidad significativa de resultados. En caso afirmativo, se toma la detección como válida. De lo contrario, se asume que el texto carece de sentido o relevancia y es descartado.

  En primer lugar, para hacer uso de esta herramienta, se debe crear el motor de búsqueda a utilizar. Esto se refiere a la selección de los sitios web en donde se

realizarán las búsquedas requeridas. Por ejemplo, si se crea un motor con los sitios `www.google.com`, `www.facebook.com` y `www.twitter.com`, las búsquedas realizadas retornarán únicamente los resultados contenidos en los sitios mencionados.

En esta solución se implementan dos motores de búsqueda, uno para el idioma inglés y otro para español. En ellos, se agregan páginas web de distintas características para intentar abarcar la mayor cantidad de resultados. Algunos de los sitios incluidos son, Wikipedia, Twitter, Facebook y Google. El resto de los sitios presentes en los motores de búsqueda definidos se listan en el apéndice B.

Una vez definidos, los motores son utilizados por *Google Custom Search* como fuente de información para hacer las búsquedas necesarias. Para la comunicación con esta herramienta se utiliza la librería *requests* [49] de Python, que permite realizar solicitudes HTTP a cualquier servidor. En este caso, se debe hacer una solicitud HTTP GET, en la que se incluya el identificador del motor de búsqueda en el que se llevarán a cabo las consultas.

La respuesta obtenida incluye el campo `searchInformation`, dentro del cual se encuentra otro, llamado `totalResults`. Este último indica la cantidad de resultados obtenidos en la búsqueda, y es el utilizado para determinar si el texto detectado será o no incluido en el índice. La cantidad mínima de resultados necesarios para el filtrado, fue definida en base a pruebas. Se realizaron una serie de búsquedas a través de la herramienta y se observó la cantidad de resultados obtenidos para textos con y sin sentido. Para los videos procesados en inglés, se determinó que el mínimo de resultados aceptado debe ser de 200.000, mientras que para español de 100.000.

Por otra parte, este servicio retorna (opcionalmente) la sección `spelling`, con el campo `correctedQuery`. Cuando dicho campo es visible quiere decir que se está aplicando la funcionalidad de "quizás quiso decir", donde el texto original de la búsqueda fue modificado por probables fallos en los datos de entrada. Esto permite corregir hitos cuyas etiquetas estén cercanas a ser una frase o palabra reconocida por el buscador.

Si bien esta solución aporta varios beneficios, cuenta con dos desventajas, un elevado precio y tiempos de respuesta largos. Para contrarrestar estos inconvenientes se busca invocar el servicio la menor cantidad de veces posible. Con este fin, se comenzaron a persistir en Firebase los campos `totalResults` y `correctedQuery` de cada consulta. De esta manera, se evita utilizar este servicio dos veces con el mismo texto, ya que antes de cada solicitud se verifica si la misma no fue realizada anteriormente.

Generación y filtrado de información a partir del audio del video

Otra de las fuentes de información a explotar para la generación de índices es el audio. Los pasos necesarios para su extracción se detallan a continuación:

- **Extracción del audio del video**

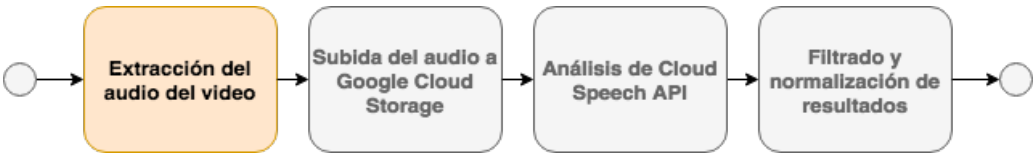


Figura 7.9: Proceso de generación de índices a partir del audio: extracción del audio del video

El primer paso luego de haber descargado el video consiste en extraer su audio a un archivo. Esto se realiza a través del utilitario de audio visto en la seccíon 7.1.2.5.

- **Subida del audio a repositorio en la nube**

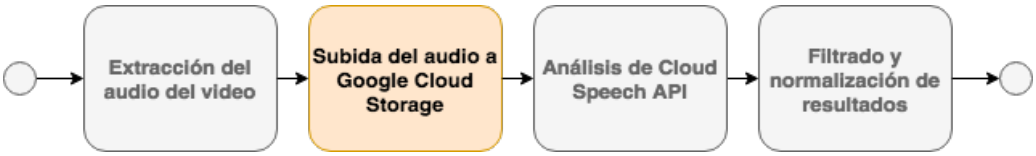


Figura 7.10: Proceso de generación de índices a partir del audio: subida del audio a Google Cloud Storage

La herramienta de procesamiento de audio que ofrece Google, requiere que el archivo se encuentre disponible en el contenedor Google Cloud Storage. La autenticación y subida de elementos es análoga al caso visto para videos e imágenes anteriores.

- **Comunicación con Google Cloud Speech API**

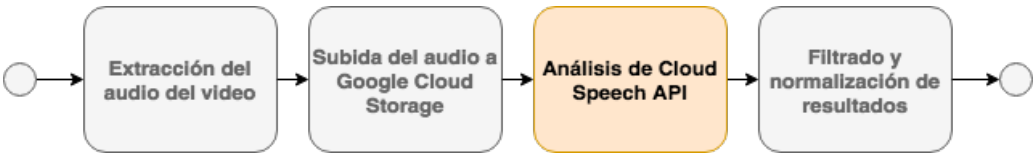


Figura 7.11: Proceso de generación de índices a partir del audio: análisis de *Google Cloud Speech API*

Con el identificador del archivo de audio dentro de Google Cloud Storage, se envía el pedido de procesamiento a *Cloud Speech API* utilizando la ya mencionada librería `google-cloud` de Python. Recibe como parámetro de entrada el lenguaje en el que se desea que el audio sea procesado. En caso que no se especifique el lenguaje, se toma por defecto el inglés.

- **Filtrado y normalización de resultados**

  Como la gran mayoría de herramientas de detección, *Cloud Speech API* retorna un nivel de confianza con cada una de sus detecciones. Este es utilizado para filtrar información que pudo haber sido producto de un error en el reconocimiento. Un detalle sobre esta herramienta es que en la respuesta aparecen solamente palabras bien formadas, esto es una ventaja ya que no es necesario crear algoritmos ni estructuras complejas para excluir palabras del índice final. Luego de filtradas las palabras que no alcanzan el nivel de confianza deseado, se estandariza la respuesta en el mismo formato definido para todos los proveedores.

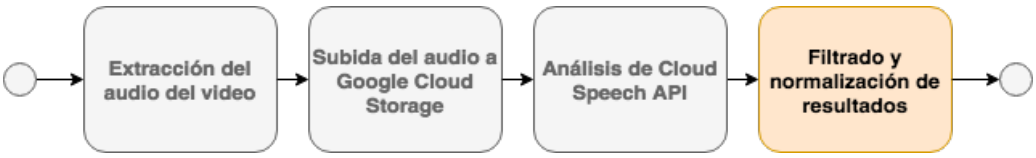


Figura 7.12: Proceso de generación de índices a partir del audio: filtrado y normalización de resultados

Unificación de información

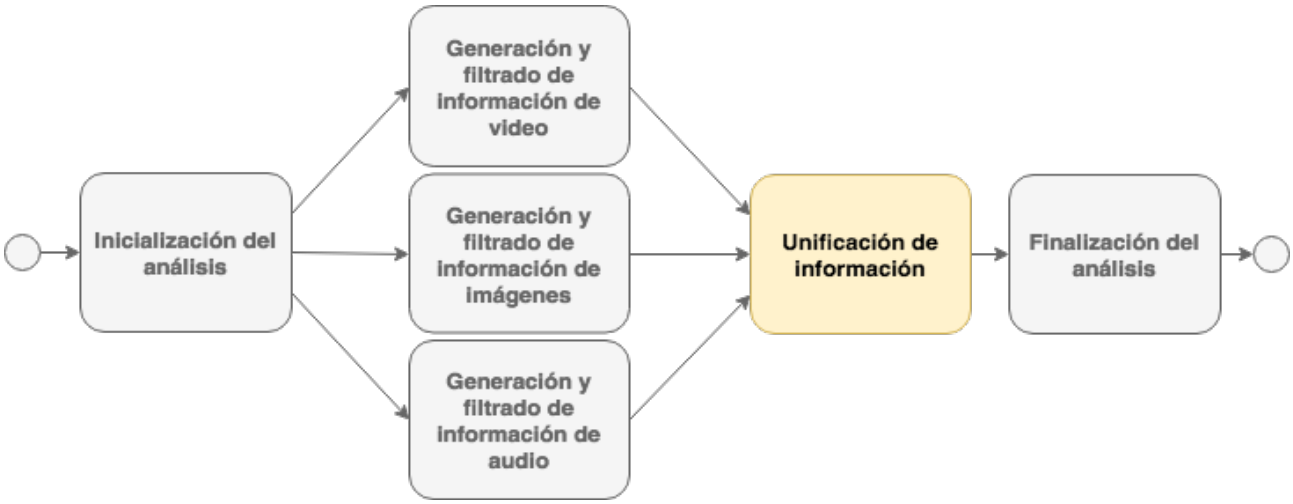


Figura 7.13: Proceso de generación de índices: unificación de información

Esta etapa del proceso comienza cuando finalizan los tres tipos de procesamientos ya mencionados. Para la unificación de la información generada se crea, en primera instancia, un mapa que contiene todos los hitos tras el procesamiento del video, de las imágenes y del audio de manera separada. Cada una de estas secciones se organiza de manera que la clave sea la etiqueta detectada y el valor una lista de los tiempos del video en los que esa entidad fue detectada.

Una vez generado este mapa, se le aplica un algoritmo que se encarga de generar el índice final del video. Este itera sobre los hitos, verificando para cada uno si los instantes de tiempo asociados se diferencian en un segundo. Si esto sucede, se consideran parte de un mismo intervalo. Luego, para estos intervalos se calcula el inicio y duración del mismo. Por ejemplo, si

para un video se tiene la etiqueta “perro” en los segundos 10, 11, 15, 20, 21 y 22, se tendrá en el índice la etiqueta perro e instantes como se ve en la siguiente tabla:

Inicio (segundos)	Duración (segundos)
10	1
15	0
20	2

Cuadro 7.2: Ejemplo de unificación de intervalos de tiempo de un índice

Finalización del análisis

Figura 7.14: Proceso de generación de índices - Finalización de análisis

Para dar como finalizado el procesamiento del video, se persiste el índice generado en la base de datos, se adapta la información a formato XML y se la envía a Videos++1. Además, para optimizar el uso del espacio en el servidor, se elimina el video, junto con sus imágenes y audio asociados, que se habían almacenado localmente.

7.1.3.2. Hitos más relevantes por categoría

Como se adelanta en el capítulo 4, sección 4.2.2.1, esta funcionalidad se implementa con el objetivo de brindar información que pueda ser de utilidad para trabajos a futuro. Se dispara cuando un usuario desde Video++1 selecciona los hitos que le resultan de interés dentro de un video y estos son enviados a Video++2 para su almacenamiento.

El servicio que recibe esta información se disponibiliza vía API en un método HTTP POST. Se envía dentro del cuerpo de la solicitud el campo `videos` que corresponde a una lista de objetos compuestos por la URL del video y la etiqueta seleccionada como relevante para ese video. Por ejemplo:

```
{
    "videos": [
        {
            "url": "https://www.youtube.com/watch?v=5mvpS-yvij0",
            "tag": "gol"
        }
    ]
}
```

Este procedimiento, recorre los objetos obteniendo la categoría del video correspondiente, previamente almacenada en Firebase. Luego, se procede a agregar la etiqueta enviada junto al video a la lista de palabras relevantes de la categoría asociada al mismo. Cada etiqueta dentro de una categoría mantiene un registro de la cantidad de veces que fue marcada como interesante. El campo en la base de datos se llama `times_liked`, y es inicializado con el valor 1 cuando se recibe el primer "me gusta" sobre esa etiqueta en dicha categoría. En caso de que la entidad que se desea agregar ya se encuentre en el listado, se incrementa en una unidad el campo `times_liked` mencionado.

7.2. Front-end

En esta sección se describe la implementación del módulo front-end de la solución desarrollada, profundizando en los puntos técnicos más interesantes.

7.2.1. Tecnologías y lineamientos seguidos

La aplicación front-end fue implementada utilizando el framework de JavaScript: Angular 5. Además de ya contar con experiencia previa utilizando esta tecnología, la decisión se basa en que es una de las más avanzadas hasta la fecha para desarrollos de este tipo [53] [3]. Adicionalmente, se siguieron los lineamientos dictaminados por las normativas de diseño *Material Design* [40], con el objetivo de ofrecer una interfaz agradable para el usuario. En la figura 7.15 se puede observar la interfaz implementada.

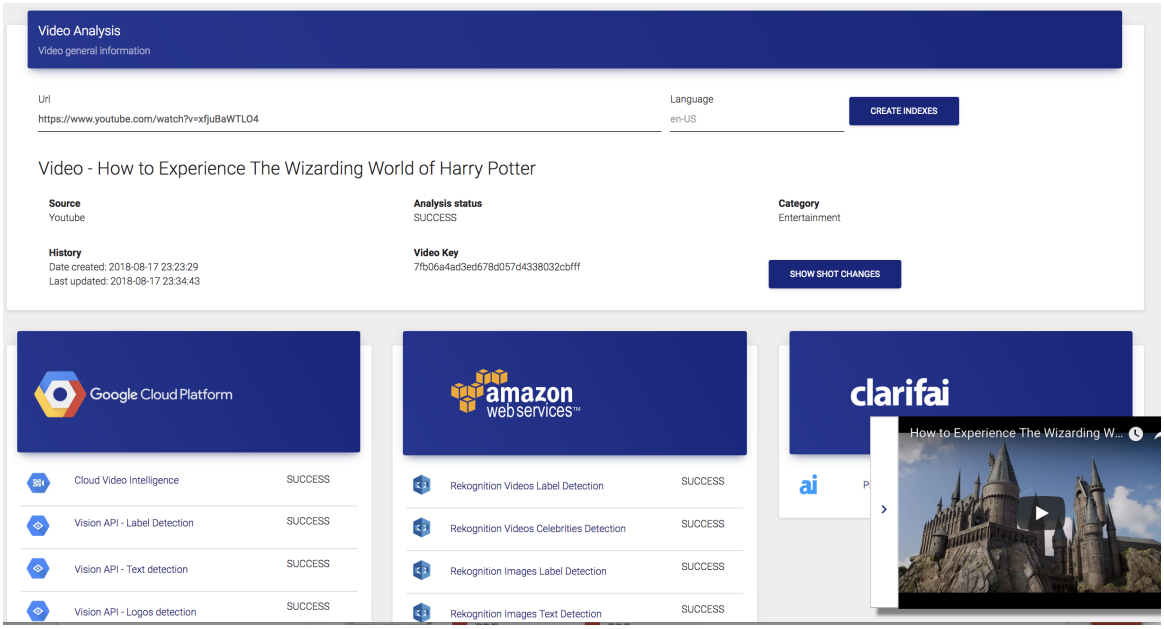


Figura 7.15: Módulo de front-end diseñado.

7.2.2. Patrones de diseño

7.2.2.1. Modelo Vista Controlador

La arquitectura del front-end se apoya en el patrón de Modelo Vista Controlador (MVC). Este se compone de tres piezas fundamentales:

- **Modelo:** Es la representación de la información utilizada por el sistema. Gestiona tanto las consultas como actualizaciones de los datos. En este caso, se trata de la información persistida en Firebase.
- **Vista:** Presenta el modelo en un formato comprensible y amigable para el usuario. Sirve de interfaz a través de la cual se interactúa con el sistema. Este módulo, está compuesto por los archivos HTML junto con sus estilos CSS.
- **Controlador:** Responde a eventos, generalmente acciones del usuario, e invoca peticiones al modelo cuando se hace alguna solicitud sobre los datos. A su vez, puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta el modelo. Por lo tanto, el controlador oficia de intermediario entre la vista y el modelo.

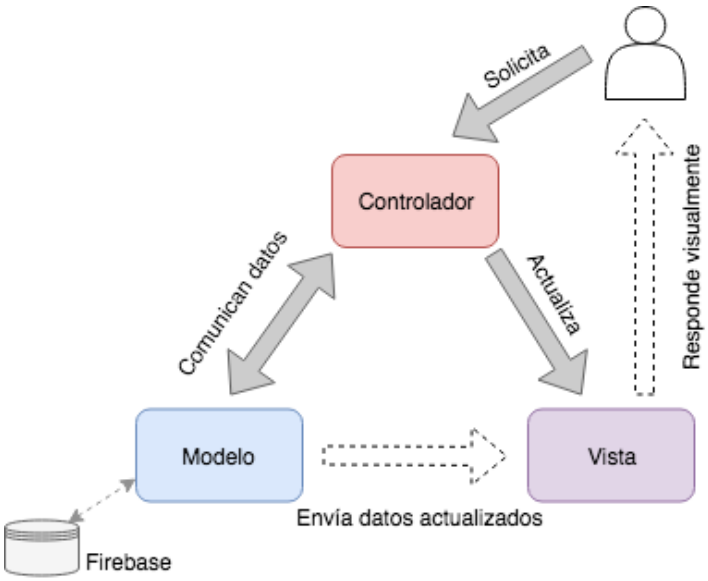


Figura 7.16: Representación gráfica del patrón MVC

7.2.2.2. Observador

El patrón Observador define una dependencia entre objetos, de manera que cuando uno cambia de estado, lo notifica a todos los dependientes. Estos últimos son llamados "observadores" ya que, de cierta manera, monitorean el estado del objeto del cual desean estar enterados.

Este comportamiento se ve reflejado a la hora de desplegar información en la interfaz web. En este caso particular, el observador estaría representado por la aplicación front-end, monitoreando los datos almacenados en Firebase. De esta manera, si ocurre algún cambio en los datos, esto se refleja de manera automática en la interfaz de usuario. Esta actualización se hace posible gracias a la funcionalidad de notificaciones de la base de datos (ver capítulo 5, sección 5.4).

7.2.3. Módulo de autenticación

Este módulo se implementó utilizando el servicio Firebase Authentication. Este se integra de forma sencilla al proyecto y brinda la posibilidad de impedir que usuarios nos deseados accedan a la aplicación. Para esto, se agregó un formulario de autenticación en el cual se solicita un usuario y una contraseña válidos.

7.2.4. Ambiente productivo

Para lograr una aplicación disponible a través de un navegador web, se optó por alojar la aplicación desarrollada en un servidor web. Para esto, se utilizó el servicio Firebase Hosting, que ofrece de forma gratuita la posibilidad de alojar la aplicación web en la nube de Googley ser accedida a través de una URL.

7.3. Persistencia de datos

Como se mencionó previamente, la base de datos seleccionada fue Firebase. Esta es utilizada tanto por la aplicación front-end, como la back-end.

Comunicación con la aplicación de back-end

Para lograr la comunicación entre el back-end y la base de datos, se utilizó la librería de Python `python-firebase` versión 1.2.

Comunicación con la aplicación de front-end

Para la comunicación entre Firebase y el front-end, se utilizó la librería `AngularFire`.

Capítulo 8

Pruebas realizadas

En la presente sección se describen las diferentes pruebas que se realizaron sobre el sistema con el fin de validar que el funcionamiento sea el esperado. Estas se dividen según las distintas iteraciones, donde se plantean los objetivos y las conclusiones de cada una.

8.1. Iteración 1: normalización de las respuestas

8.1.1. Objetivos

En la primer iteración del proyecto se planteó como objetivo conectar el sistema con las distintas herramientas seleccionadas (ver capítulo 3). Al final de esta etapa, la información generada por cada uno de estos servicios debería ser almacenada en Firebase en el formato establecido en el modelo de datos.

8.1.2. Pruebas realizadas

La evaluación del prototipo en esta fase consistió en verificar que lo que se estuviera guardando en la base de datos fuera correcto. Esto quiere decir, que se correspondiera con la información retornada por las herramientas y que el modelo de datos definido se respetara.

8.1.3. Conclusiones

Los objetivos establecidos para la iteración fueron alcanzados. Al momento de procesar un video, la información obtenida era normalizada y persistida en la base de datos en el formato esperado.

8.2. Iteración 2: unificación de las respuestas

8.2.1. Objetivos

En esta etapa se plantean, por un lado, los objetivos fundamentales para la generación de los índices y, por otro, algunos de carácter complementario para alcanzar una solución de mayor valor.

8.2.1.1. Objetivos principales

- **Unificación de respuestas**: guardar en la base de datos, el índice combinando los hitos detectados por los distintos servicios de inteligencia artificial.
- **Comunicación con Video++1**: al finalizar la generación del índice, el sistema debe notificar a Video++1, incluyendo los hitos detectados para el video en dicho mensaje.

8.2.1.2. Objetivos complementarios

- **Filtrado por confianza**: desarrollar el mecanismo de filtrado por nivel de confianza fijando, en esta primera versión, un valor para cada herramienta para ser ajustado en posteriores iteraciones según los resultados obtenidos.
- **Visualización de resultados**: construir el front-end donde visualizar los hitos detectados por las distintas herramientas, así como el índice final generado.

8.2.2. Pruebas realizadas

Para verificar los resultados obtenidos en esta iteración, se realizaron pruebas funcionales y de integración. Se dividieron en los siguientes casos:

- **Verificar unificación de índices**: se observó si el índice resultante tenía coherencia con la información generada por las herramientas de inteligencia artificial.
- **Pruebas de integración**: se validó con el equipo de Video++1 que las solicitudes realizadas al sistema recibían la notificación esperada con el formato especificado.
- **Verificación de filtro por confianza**: se chequeó el nivel de confianza de los hitos generados, superaran el límite establecido en esta etapa.
- **Análisis de calidad del índice**: se definió un conjunto de videos que abarquen la mayor cantidad de categorías existentes en el sistema y se observaron los índices generados. Para su evaluación, se utilizaron los criterios definidos en el capítulo 3. Algunos aspectos incluidos en estos criterios son, la generación de falsos positivos, elementos existentes pero no detectados y el grado de detalle de los resultados.

8.2.3. Conclusiones

Se cumplieron los objetivos planteados para esta iteración. Sin embargo, al momento de evaluar la calidad de los índices, se encontraron dos grandes problemas. El primero fue la exorbitante cantidad de hitos que se generaban para un mismo vídeo. El segundo y más importante, la existencia de etiquetas formadas por cadenas de caracteres sin sentido o que conformaban palabras irrelevantes. Estos resultados dieron dio pie al comienzo de una nueva iteración.

8.3. Iteración 3: mejora de la calidad del índice

8.3.1. Objetivos

Elevar la calidad de los índices generados a través de la aplicación de las siguientes estrategias:

- **Ajuste de niveles de confianza**: en esta iteración se modificaron los niveles de confianza aceptados para cada herramienta. Dichos valores surgen de la observación de los resultados obtenidos. El listado de herramientas con sus respectivos umbrales se encuentra en el capítlo 5, cuadro 5.1.
- **Filtrado de texto incorrecto**: fueron filtradas las etiquetas con cadenas de caracteres que no representan palabras o que no tengan sentido. Dicha tarea se llevó a cabo mediante el uso de diccionarios y la cantidad de resultados obtenidos a través de la herramienta *Google Custom Search*.
- **Filtrado de texto irrelevante**: se filtraron palabras que no aportan valor al índice generado. Se utiliza un listado para el idioma español y otro para el inglés. [13] [52]

8.3.2. Pruebas realizadas

Para esta iteración se utilizaron los mismos videos de la iteración anterior y se aplicaron los mismos criterios para su evaluación. De esta manera, se realizó una comparación entre los resultados obtenidos en las distintas iteraciones. Se prestó particular atención al filtrado de los hitos no deseados. En el apéndice C se pueden observar las etiquetas generadas, antes y después de aplicar las mejoras planteadas en esta iteración.

8.3.3. Conclusiones

Como se puede observar en el apéndice referenciado, se disminuyó considerablemente la cantidad de hitos con etiquetas incorrectas e irrelevantes. Además, se observa que se filtran muy pocas que sí podrían aportar valor a la solución, lo cual era un riesgo a asumir cuando se tomó la decisión de aplicar filtros. El final de esta iteración marca la terminación de la construcción del prototipo.

Capítulo 9

Gestión del proyecto

9.1. Descripción

El presente proyecto tuvo sus comienzos en el mes de setiembre del 2017. En ese momento se llevó a cabo la primera reunión entre los profesores e integrantes del equipo que sirvió de introducción a la temática que se deseaba abordar. Luego del primer contacto, se comenzó con el estudio de herramientas de inteligencia artificial existentes, para posteriormente, iniciar la implementación del prototipo y la redacción del informe.

La organización en cuanto a formato de trabajo, se realizó mediante jornadas presenciales dentro del grupo de estudiantes y reuniones quincenales con los tutores del proyecto. Durante los primeros seis meses, la frecuencia de reuniones dentro del equipo fue de tres por semana en promedio, mientras que en los siguientes cuatro meses el número ascendió a cuatro y en los últimos dos, a cinco.

9.2. Metodología de desarrollo y plan de trabajo

Desde un principio se planteó realizar la documentación del proyecto en paralelo al desarrollo del prototipo de software. El objetivo que persigue esta decisión es contar con una fuente de información que respalde las investigaciones realizadas evitando reiterar tareas.

A continuación se presenta el plan de trabajo planteado inicialmente con las modificaciones generadas en el proceso:

1. Análisis de requerimientos del sistema.
2. Estudio de herramientas y selección de las mismas.
3. Determinación del alcance y diseño de la solución. Selección de tecnologías a utilizar.
4. Implementación y testing:
    a) Iteración 1: establecer comunicación con los distintos servicios de inteligencia artificial y normalización del formato de las respuestas.

b) Iteración 2: unificación de información obtenida por las distintas herramientas para disponibilizar índices vía API. Integración con proyecto Video++1. Fin de la primer versión del prototipo.

c) Análisis del prototipo en base a pruebas y reuniones con tutores. Definición de segunda versión del sistema a construir.

d) Iteración 3: construcción de la versión resultante del análisis del prototipo liberado en la iteración anterior. En este punto surgen requerimientos que no estaban previstas en el plan inicial, el objetivo de esta etapa es mejorar la calidad de los índices.

e) Pruebas generales del sistema, correcciones necesarias.

5. Culminación de la documentación del trabajo.

En todas las etapas del proceso se definieron tareas a resolver y se dispusieron responsables. Al inicio de cada jornada de trabajo se disponía un tiempo para comentar sobre el avance y los problemas detectados que podían ser bloqueantes.

A continuación se presentan dos gráficos: en la figura 9.1 se muestra la distribución de los integrantes del equipo dentro de los distintos tipos de tareas en cada mes. Mientras que en la figura 9.2 se da un resumen general del reparto de esfuerzo durante todo el proyecto.

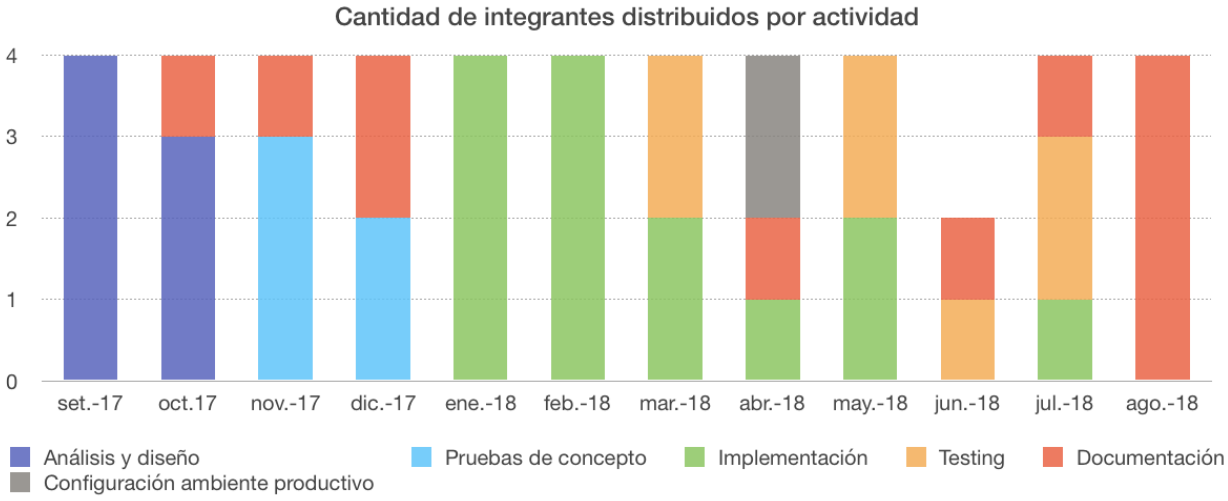


Figura 9.1: Distribución de la cantidad de personas asignadas por tipo de tarea durante cada mes

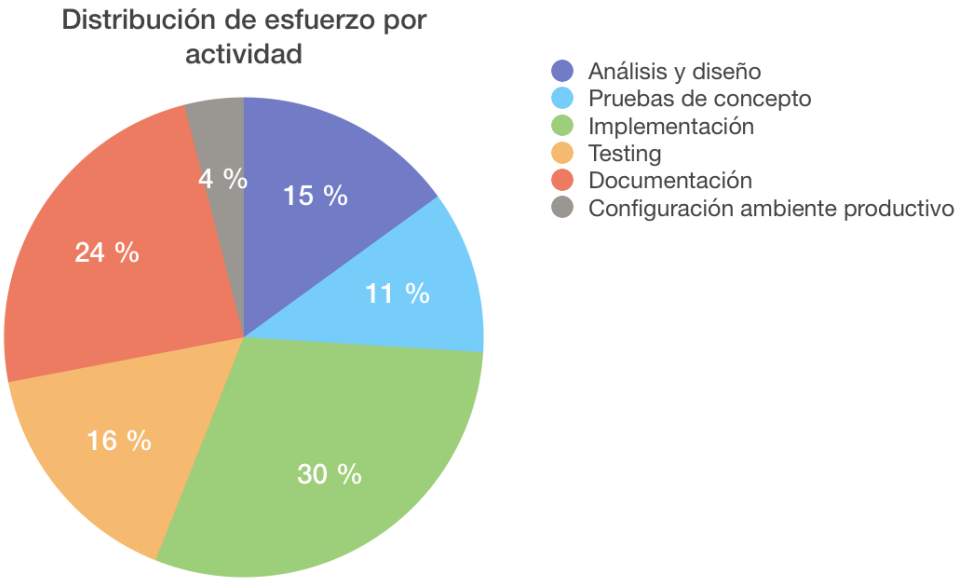


Figura 9.2: Distribución del esfuerzo por tipo de tarea durante todo el proyecto

Un aspecto a tener en cuenta sobre la gráfica de distribución mensual, es que una persona no siempre se dedicaba un mes entero al mismo tipo de tarea. Sin embargo, en lineas generales la distribución mostrada en la gráfica se cumple ya sea por dedicación completa a una clase de tarea o por la rotación de los integrantes dentro de las mismas.

Otro aspecto a destacar es que durante el mes de junio dos de los integrantes del grupo no estuvieron presentes por motivo de viaje, disminuyendo la cantidad de recursos disponibles.

Finalmente, un detalle importante es que, excluyendo la configuración del ambiente productivo, todos los participantes estuvieron involucrados en todos los tipos de tareas.

9.3. Hitos importantes del proyecto

En esta sección se listan los hitos más relevantes durante el desarrollo del proyecto:

- 16/08/2017: Primera reunión con los tutores.
- 08/11/2017: Respuestas positivas por parte de Google y Clarifai sobre pedido de crédito para uso de sus herramientas.
- 12/12/2017: Primera definición del prototipo de software a construir.
- 14/02/2018: Fin de la primer iteración que incluye el prototipo conectado a todos los proveedores y almacenado de información en base de datos.
- 15/03/2018: Primera reunión con los integrantes de Video++1.
- 05/04/2018: Fin de la segunda iteración donde se encuentran unificandas la respuestas de todos los proveedores.

- 20/04/2018: Primera liberación a producción a través de Google App Engine. Se expone servicio público para interactuar con el sistema.
- 25/04/2018: Definición de la segunda iteración del prototipo.
- 28/04/2018: Primeros capítulos completos de la documentación.
- 08/05/2018: Primera interacción entre sistemas Video++1 y Video++2.
- 28/05/2018: Desarrollo completo de la segunda iteración del prototipo.
- 27/07/2018: Versión final del software luego de correcciones posteriores a fase de pruebas.
- 31/08/2018: Primera versión de la documentación completa.
- 05/09/2018: Última versión de la documentación.

9.4. Cronograma

A continuación se describe el cronograma llevado a cabo a lo largo del proyecto:

- **Setiembre - Octubre 2017**: Introducción a los proyectos Video++1 y Video++2, definición de la problemática, primer aproximación del alcance.
- **Noviembre - Diciembre 2017**: Pruebas de concepto de las diferentes herramientas. Comunicación con Google, Amazon, Clarifai y Microsoft.
- **Enero - Marzo 2018**: Construcción de la primer versión del prototipo de software. Incluye generación de índices a través de información obtenida por todas las herramientas seleccionadas.
- **Abril 2018**: Liberación del prototipo en ambiente productivo, sistema disponible públicamente. Se construye aplicación web para visualización de los índices generados por la herramienta y la unificación de los mismos en un índice final.
- **Mayo 2018**: Definición y construcción de segunda iteración de desarrollo. El prototipo liberado cuenta con algoritmos de filtrado para generar un índice de mayor calidad. Se finalizan primeras versiones de los capítulos iniciales del informe.
- **Junio - Agosto 2018**: Pruebas generales del sistema, retoques del prototipo. Etapa final de la documentación.
- **Setiembre 2018**: Se termina la documentación del trabajo.

Capítulo 10

Conclusiones y trabajo a futuro

En este capítulo se presentan las conclusiones del trabajo realizado. También se exponen las dificultades atravesadas durante el mismo y las principales líneas de trabajo a futuro.

10.1. Conclusiones generales

En primer lugar, se concluye que los objetivos planteados al principio del proyecto fueron cumplidos. Se alcanzó un prototipo de software que satisface los requerimientos deseados, buscando, incluso, formas de mejorar la calidad del producto final. Como aspecto primordial a destacar, se lograron generar índices de manera automática para videos. Situación que era una gran incógnita a inicios del trabajo.

Otro de los objetivos, que resultó ser el factor determinante para la generación automática de índices, fue la conectividad con un amplio espectro de herramientas de inteligencia artificial. En este punto, se destaca que se encontró la manera de analizar las tres fuentes más significativas de información que se pueden adquirir de un video, como lo son el propio video, imágenes extraídas del mismo y su audio. En este sentido, se llevó a cabo una profunda evaluación de las herramientas existentes, explorando sus características y divisando dónde cada una de ellas logra aportar valor a la funcionalidad de generación de índices. Además de interactuar con un gran número de servicios ofrecidos por las empresas de vanguardia en el desarrollo de software en la actualidad, el prototipo implementado se integra con la aplicación Videos++1. El contar con una aplicación de front-end que permita visualizar los índices generados y trabajar sobre ellos aporta gran valor a la solución conjunta.

En tanto a la calidad de la información generada, se destaca que se elaboraron mecanismos que buscan aprovechar, de manera directa o indirecta, la intervención del humano para aumentar el valor de los hitos involucrados en cada índice. Gracias a estas estrategias, se visualiza una clara mejora en las comparaciones de los índices en las distintas iteraciones del prototipo. De todos modos, dentro del equipo queda la expectativa de producir resultados con etiquetas más específicas, a través de mecanismos de mayor complejidad que se especialicen dentro de distintas categorías de videos. En este punto, se plantea la discusión de qué índices son importantes para quién, y si con el afán de filtrar información irrelevante se quitan hitos que para

los usuarios son realmente útiles. Dentro de este trabajo, la postura adoptada fue aumentar la calidad de los índices eliminando la información que se puede catalogar como incorrecta, dejando la puerta abierta a los usuarios y a nuevos algoritmos en trabajos futuros de elaborar índices con un grado de precisión más alto.

En líneas generales, se llega a la conclusión de que el proyecto fue llevado a cabo de forma exitosa en tiempos adecuados, donde los niveles de aprendizaje en cuanto a tecnologías de carácter innovador es elevado. Además, el prototipo de software construido cumple con los requisitos que se plantearon inicialmente y se detecta un gran potencial para su evolución a futuro.

10.2. Dificultades encontradas

10.2.1. Solicitud de créditos

A principio del proyecto se obtuvieron respuestas que alentaban a obtener financiamiento por parte de Amazon, Google y Clarifai. Afortunadamente, durante el desarrollo del prototipo se dieron las aprobaciones de crédito para las herramientas de los últimos dos. El crédito otorgado por Google dio la posibilidad de utilizar una variedad importante de servicios. Además, fue clave para montar la infraestructura de hardware necesaria para satisfacer los requisitos de cómputo del sistema.

En el caso de Amazon, se inició la solicitud pero la respuesta afirmativa no llegó antes de la finalización del proyecto. Sin embargo, se pudo hacer uso de sus herramientas con el crédito gratuito para pruebas.

10.2.2. Costo de herramientas

El hecho de que todas las herramientas de inteligencia artificial o servicios consumidos, no sean gratuitos, creó la necesidad de usarlos con responsabilidad para no malgastar los créditos de prueba disponibles. Si bien se obtuvo financiación para las herramientas de Google y Clarifai, estas no fueron efectuadas hasta una etapa avanzada del prototipo. Por el lado de Amazon, se logró llegar a final del proyecto utilizando los servicios de prueba. Algunos ejemplos de acciones para evitar el uso masivo de las herramientas son, el recorte de imágenes en instantes clave de los videos, el guardado del historial de búsquedas realizadas a Google Custom Search para no repetir pedidos idénticos y la aplicación escalonada de filtros dentro de un índice desde los más simples a los más costosos.

10.2.3. Tiempos de procesamiento

El prototipo tiene alta dependencia de sistemas externos que realizan tareas de procesamiento pesadas. A su vez, las descargas de videos y análisis de los mismos dentro de la aplicación requieren de una cantidad de tiempo que no es despreciable. A esto se le debe sumar que el

mecanismo de generación de índices se puede disparar de manera simultánea para varios videos, situación que sucedió en instancias de pruebas. Por lo tanto, fue necesario el desarrollo de estrategias que eviten conexiones a sistemas externos de forma innecesaria. En cuanto a la capacidad de cómputo interna, el haber obtenido crédito por parte de Google permitió montar una estructura con escalamiento automático que aumente la cantidad de pedidos simultáneos soportados por el sistema.

10.3. Trabajo a futuro

En esta sección se describen las posibles mejoras al prototipo desarrollado para dotarlo de un mayor conjunto de funcionalidades y aumentar la calidad de los índices generados.

Explotar la retroalimentación de los usuarios

Aprovechar la funcionalidad desarrollada, que permite construir un listado ponderado de etiquetas por categoría, mediante la implementación de algoritmos que tomen estos datos de entrada y decidan las mejores etiquetas para agregar a los índices.

Buscador de Google por categorías

Como se mencionó en el capítulo 7, sección 7.1.3.1, el buscador de Google es utilizado para saber si una etiqueta cuenta con un número de resultados considerable en una búsqueda, dando una medida de qué tanto valor puede tener para un usuario. Dichos resultados se obtienen de una base de sitios web, que en la versión liberada del prototipo, cuenta con un conjunto variado pero limitado de páginas. En esta propuesta a futuro, se pretende generar motores de búsqueda por categorías de videos. De esta manera se podría buscar la etiqueta en un conjunto de sitios más específicos del contexto al que el video pertenece, pudiendo obtener mejores resultados.

Soporte de traducciones

El prototipo actual soporta los lenguajes inglés y español. Además, las herramientas de inteligencia artificial devuelven la información de etiquetas en inglés. Una funcionalidad que aportaría valor al sistema sería procesar datos de entrada en una variedad de idiomas y poder devolver la respuesta en el idioma especificado por el usuario.

Detectar cambios en los decibeles del audio

Medir los niveles del volumen en el audio extraído del video, con el objetivo de reconocer momentos interesantes dentro del mismo. Ejemplos de estos pueden ser, explosiones, gritos o

música. Luego, prestar particular atención a esos instantes, por ejemplo, tomando capturas del video y procesándolas con las herramientas de análisis de imágenes.

Filtros compuestos por proximidad

En ocasiones se detectan entidades en intervalos muy cercanos de tiempo que podrían estar relacionadas. Sería de interés encontrar dichas asociaciones y generar etiquetas compuestas que otorguen mayor especificidad. Por ejemplo, si se detecta la etiqueta "gol" y el apellido "Suárez" con una diferencia muy pequeña de tiempo, puede tratarse de un "gol de Suárez" y retornar esa etiqueta en lugar de dos separadas sería más valioso para el usuario.

Versionado de índices

Contar con un sistema de versionado de los índices generados permitiría realizar una evaluación de la evolución de la calidad de la información que el sistema produce.

Categorizador de videos

La solución actual utiliza las categorías de videos proporcionadas por la plataforma YouTube. Crear un categorizador propio permitiría explotar los beneficios de la contextualización de los videos para contenidos de cualquier procedencia.

Aumentar la cantidad de plataformas de video soportadas

El presente prototipo soporta el procesamiento de videos a través de una URL de YouTube o de la ruta del propio video dentro de internet. Se pretende que futuras versiones puedan realizar análisis directamente sobre plataformas como Vimeo, Dailymotion, OpenFing, entre otras.

Soportar videos más largos

Aumentar la restricción de la longitud de videos permitidos por el sistema. Esto se puede realizar cortando los videos cada 30 minutos y analizarlos por separado.

Apéndice A

Ejemplos de análisis de las herramientas

A continuación, se ilustran algunos fragmentos de las respuestas obtenidas por las herramientas durante pruebas realizadas para su evaluación. Las mismas, se agrupan por el tipo de dato que toman como entrada para elaborar el análisis.

A.1. Resultados de análisis de herramientas de video

A.1.1. Detección de etiquetas

Herramientas evaluadas:

- Cloud Video Intelligence API - Label Detection
- Rekognition Video Features - Label Detection
- Clarifai Predict Videos

Se realiza un análisis para el video "Coca Cola Commercial - First Kiss" [55]
Respuesta obtenida por Cloud Video Intelligence - Label Detection:


```
{
    "labelAnnotations": [
          {
            "categoryEntities": [
              {
                "description": "people",
                "entityId": "/m/09g5pq",
                "languageCode": "en-US"
              }
            ],
            "entity": {
              "description": "family",
```

```
            "entityId": "/m/09dhj",
            "languageCode": "en-US"
          },
          "segments": [
            {
              "confidence": 0.5209441,
              "segment": {
                "endTimeOffset": "7.640s",
                "startTimeOffset": "5.960s"
              }
            }
          ]
        },
        {
          "entity": {
            "description": "social group",
            "entityId": "/m/01b4q9",
            "languageCode": "en-US"
          },
          "segments": [
            {
              "confidence": 0.76379156,
              "segment": {
                "endTimeOffset": "3.040s",
                "startTimeOffset": "1.960s"
              }
            },
            {
              "confidence": 0.82779354,
              "segment": {
                "endTimeOffset": "28.960s",
                "startTimeOffset": "27.760s"
              }
            },
            {
              "confidence": 0.9146881,
              "segment": {
                "endTimeOffset": "31.160s",
                "startTimeOffset": "30.040s"
              }
            },
            {
              "confidence": 0.71292716,
              "segment": {
                "endTimeOffset": "42.680s",
                "startTimeOffset": "42.120s"
              }
            }
```

```
            ]
          },
          {
            "categoryEntities": [
              {
                "description": "vehicle",
                "entityId": "/m/07yv9",
                "languageCode": "en-US"
              }
            ],
            "entity": {
              "description": "land vehicle",
              "entityId": "/m/01prls",
              "languageCode": "en-US"
            },
            "segments": [
              {
                "confidence": 0.6298285,
                "segment": {
                  "endTimeOffset": "16.960s",
                  "startTimeOffset": "15.680s"
                }
              }
            ]
          } (...)
]
}
```


Respuesta obtenida por Rekognition Video Features - Label Detection:


```
{
 "Labels": [
      {
         "Label": {
            "Confidence": 54.4277,
            "Name": "Animal"
         },
         "Timestamp": 0
      },
      {
         "Label": {
            "Confidence": 54.372902,
            "Name": "Female"
         },
         "Timestamp": 1200
      },
      {
         "Label": {
            "Confidence": 54.372902,
```

```
            "Name": "Girl"
         },
         "Timestamp": 1200
      },
      {
         "Label": {
            "Confidence": 99.0794,
            "Name": "Human"
         },
         "Timestamp": 1200
      },
      {
         "Label": {
            "Confidence": 56.4003,
            "Name": "Lighting"
         },
         "Timestamp": 1200
      },
      {
         "Label": {
            "Confidence": 98.1588,
            "Name": "People"
         },
         "Timestamp": 1200
      },
      {
         "Label": {
            "Confidence": 67.180405,
            "Name": "Female"
         },
         "Timestamp": 1400
      },
      {
         "Label": {
            "Confidence": 67.180405,
            "Name": "Girl"
         },
         "Timestamp": 1400
      },
      {
         "Label": {
            "Confidence": 99.1127,
            "Name": "Human"
         },
         "Timestamp": 1400
      }
      (...)
}
```

Respuesta obtenida de Clarifai Predict Video:


```
{
"frames": [
                {
                   "frame_info": {
                      "index": 0,
                      "time": 0
                   },
                   "data": {
                      "concepts": [
                         {
                            "id": "ai_786Zr311",
                            "name": "no person",
                            "value": 0.98858035,
                            "app_id": "main"
                         },
                         {
                            "id": "ai_5Kp5FMJw",
                            "name": "still life",
                            "value": 0.95767486,
                            "app_id": "main"
                         },
                         {
                            "id": "ai_2gmKZLxp",
                            "name": "cold",
                            "value": 0.9310192,
                            "app_id": "main"
                         },
                         {
                            "id": "ai_tBcWlsCp",
                            "name": "nature",
                            "value": 0.91070974,
                            "app_id": "main"
                         },
                         {
                            "id": "ai_Pf2b7clG",
                            "name": "indoors",
                            "value": 0.89467424,
                            "app_id": "main"
                         },
                         {
                            "id": "ai_3PlgVmlN",
                            "name": "food",
                            "value": 0.8936278,
                            "app_id": "main"
                         },
                         {
                            "id": "ai_FsT0Zqdb",
```

```
                        "name": "summer",
                        "value": 0.8857873,
                        "app_id": "main"
                     },
                     {
                        "id": "ai_l4WckcJN",
                        "name": "blur",
                        "value": 0.8838802,
                        "app_id": "main"
                     },
                     {
                        "id": "ai_DSCFH6f6",
                        "name": "sand",
                        "value": 0.8388915,
                        "app_id": "main"
                     },
                     {
                        "id": "ai_NTTFwSHB",
                        "name": "wet",
                        "value": 0.83527887,
                        "app_id": "main"
                     },
                  ]
               }
            },
            {
               "frame_info": {
                  "index": 1,
                  "time": 1000
               },
               "data": {
                  "concepts": [
                     {
                        "id": "ai_l8TKp2h5",
                        "name": "people",
                        "value": 0.94247234,
                        "app_id": "main"
                     },
                     {
                        "id": "ai_RQccV41p",
                        "name": "woman",
                        "value": 0.94069004,
                        "app_id": "main"
                     },
                     {
                        "id": "ai_tBcWlsCp",
                        "name": "nature",
                        "value": 0.87726146,
```

```
                    "app_id": "main"
                },
                {
                    "id": "ai_xQbBc1zb",
                    "name": "girl",
                    "value": 0.87270564,
                    "app_id": "main"
                },
                {
                    "id": "ai_V2F286Wn",
                    "name": "fun",
                    "value": 0.7447124,
                    "app_id": "main"
                },
                {
                    "id": "ai_zJx6RbxW",
                    "name": "drink",
                    "value": 0.7294805,
                    "app_id": "main"
                },
                {
                    "id": "ai_Pf2b7clG",
                    "name": "indoors",
                    "value": 0.6994981,
                    "app_id": "main"
                },
                {
                    "id": "ai_4sJLn6nX",
                    "name": "dark",
                    "value": 0.683153,
                    "app_id": "main"
                },
                {
                    "id": "ai_ZSKpCCHD",
                    "name": "vertical",
                    "value": 0.64160466,
                    "app_id": "main"
                },
                {
                    "id": "ai_n9vjC1jB",
                    "name": "light",
                    "value": 0.6404547,
                    "app_id": "main"
                },
                {
                    "id": "ai_2FHPKk9B",
                    "name": "winter",
                    "value": 0.64041865,
```

```
                        "app_id": "main"
                    }
                ]
            }
        },
        {
            "frame_info": {
                "index": 2,
                "time": 2000
            },
            "data": {
                "concepts": [...]
             }
           }
    ]
}
```


A.1.2. Detección de celebridades

Herramientas evaluadas:

- Rekognition Video Features - Celebrity Detection

Se realiza un análisis para el video: Maroon 5 - Girls Like You [57]
Respuesta obtenida de Rekognition Video Features - Celebrity Detection:


```
{
  "Celebrities": [
      {
         "Celebrity":{
            "BoundingBox":{
               "Height":0.605555534362793,
               "Left":0.02421874925494194,
               "Top":0.3861111104488373,
               "Width":0.3414062559604645
            },
            "Confidence":84,
            "Id":"4aD7HD8R",
            "Name":"Ellen DeGeneres",
            "Urls":[
               "www.imdb.com/name/nm0001122"
            ]
         },
         "Timestamp":10040
      },
       {
         "Celebrity":{
            "BoundingBox":{...},
```

```
            "Confidence":84,
            "Id":"4aD7HD8R",
            "Name":"Ellen DeGeneres",
            "Urls":[
               "www.imdb.com/name/nm0001122"
            ]
         },
         "Timestamp":10120
      },
       {
         "Celebrity":{
            "BoundingBox":{...},
            "Confidence":90,
            "Id":"59D7oP8R",
            "Name":"Adam Levine",
            "Urls":[
               "www.imdb.com/name/nm1747013"
            ]
         },
         "Timestamp":10220
      }
   ]
}
```


A.1.3. Detección de contenido inapropiado

Herramientas evaluadas:

- Rekognition Video Features - Image Moderation

Respuesta obtenida de Rekognition Video Features - Image Moderation:


```
{
   "ModerationLabels": [
      {
         "ModerationLabel": {
            "Confidence": 62.899845,
            "Name": "Female Swimwear Or Underwear",
            "ParentName": "Suggestive"
         },
         "Timestamp": 4200
      },
      {
         "ModerationLabel": {
            "Confidence": 50.34825,
            "Name": "Revealing Clothes",
            "ParentName": "Suggestive"
         },
```

```
            "Timestamp": 4200
        },
        {
            "ModerationLabel": {
                "Confidence": 62.899845,
                "Name": "Suggestive",
                "ParentName": ""
            },
            "Timestamp": 4200
        },
        {
            "ModerationLabel": {
                "Confidence": 50.427277,
                "Name": "Explicit Nudity",
                "ParentName": ""
            },
            "Timestamp": 4400
        },
        {
            "ModerationLabel": {
                "Confidence": 50.427277,
                "Name": "Partial Nudity",
                "ParentName": "Explicit Nudity"
            },
            "Timestamp": 4400
        },
        {
            "ModerationLabel": {
                "Confidence": 51.105263,
                "Name": "Revealing Clothes",
                "ParentName": "Suggestive"
            },
            "Timestamp": 4400
        }
        (...)
}
```


A.2. Resultados de análisis de herramientas de imágenes

A.2.1. Detección de etiquetas

Herramientas evaluadas:

- Vision API - Label Detection
- Amazon Rekognition Images Features - Label Detection
- Clarifai Predict Images

Imagen utilizada para el análisis:

Figura A.1: Imagen seleccionada para el análisis [33]

Respuesta obtenida por Vision API - Label Detection:


```
{
 "labelAnnotations": [
    {
      "mid": "/m/0c_jw",
      "description": "sofa",
      "score": 0.9646166,
      "topicality": 0.9646166
    },
    {
      "mid": "/m/06ht1",
      "description": "room",
      "score": 0.8536701,
      "topicality": 0.8536701
    },
    {
      "mid": "/m/03ssj5",
      "description": "bed",
      "score": 0.84865886,
      "topicality": 0.84865886
    },
    {
      "mid": "/m/05kyg_",
      "description": "night",
      "score": 0.84309906,
      "topicality": 0.84309906
    },
    {
      "mid": "/m/0fqfqc",
      "description": "drawer",
      "score": 0.75277656,
```

```
      "topicality": 0.75277656
    },
    {
      "mid": "/m/0l7_8",
      "description": "floor",
      "score": 0.6597198,
      "topicality": 0.6597198
    },
    {
      "mid": "/m/02rfdq",
      "description": "interior design",
      "score": 0.653456,
      "topicality": 0.653456
    },
    {
      "mid": "/m/04mygm",
      "description": "bed sheet",
      "score": 0.6263813,
      "topicality": 0.6263813
    },
    {
      "mid": "/m/01l0mw",
      "description": "home",
      "score": 0.6073166,
      "topicality": 0.6073166
    },
    {
      "mid": "/m/052l6z",
      "description": "box spring",
      "score": 0.5110192,
      "topicality": 0.5110192
    }
  ]
}
```


Respuesta obtenida por Rekognition Image Features - Label Detection:


```
{
"Labels": [
        {
            "Name": "Flora",
            "Confidence": 99.32398986816406
        },
        {
            "Name": "Jar",
            "Confidence": 99.32398986816406
        },
        {
            "Name": "Plant",
```

```
            "Confidence": 99.32398986816406
        },
        {
            "Name": "Potted Plant",
            "Confidence": 99.32398986816406
        },
        {
            "Name": "Pottery",
            "Confidence": 99.32398986816406
        },
        {
            "Name": "Vase",
            "Confidence": 99.32398986816406
        },
        {
            "Name": "Bedroom",
            "Confidence": 86.37647247314453
        },
        {
            "Name": "Indoors",
            "Confidence": 86.37647247314453
        },
        {
            "Name": "Interior Design",
            "Confidence": 86.37647247314453
        },
        {
            "Name": "Room",
            "Confidence": 86.37647247314453
        },
        {
            "Name": "Furniture",
            "Confidence": 85.27618408203125
        },
        {
            "Name": "Mirror",
            "Confidence": 84.7614517211914
        },
        {
            "Name": "Sideboard",
            "Confidence": 81.23590087890625
        },
        {
            "Name": "Studio Couch",
            "Confidence": 53.22001647949219
        },
        {
            "Name": "Blossom",
```

```
            "Confidence": 52.82499694824219
        },
        {
            "Name": "Flower",
            "Confidence": 52.82499694824219
        },
        {
            "Name": "Flower Arrangement",
            "Confidence": 52.82499694824219
        },
        {
            "Name": "Ornament",
            "Confidence": 52.82499694824219
        },
        {
            "Name": "Dresser",
            "Confidence": 51.81643295288086
        },
        {
            "Name": "Tabletop",
            "Confidence": 51.193267822265625
        }
    ],
    "OrientationCorrection": "ROTATE_0"
}
```


Respuesta obtenida por Clarifai Predict Images:


```
{
    "concepts": [
        {
            "id": "ai_c9n7SB25",
            "name": "furniture",
            "value": 0.99918735,
            "app_id": "main"
        },
        {
            "id": "ai_pPxqdnP5",
            "name": "room",
            "value": 0.99639225,
            "app_id": "main"
        },
        {
            "id": "ai_B2TBwp8F",
            "name": "bed",
            "value": 0.98482597,
            "app_id": "main"
        },
        {
```

```
            "id": "ai_bJt1PfQ5",
            "name": "bedroom",
            "value": 0.97794116,
            "app_id": "main"
        },
        {
            "id": "ai_x3vjxJsW",
            "name": "home",
            "value": 0.97612846,
            "app_id": "main"
        },
        {
            "id": "ai_BJP7PBvG",
            "name": "apartment",
            "value": 0.97516066,
            "app_id": "main"
        },
        {
            "id": "ai_N8lt4NQt",
            "name": "pillow",
            "value": 0.97018206,
            "app_id": "main"
        },
        {
            "id": "ai_Pf2b7clG",
            "name": "indoors",
            "value": 0.9710376,
            "app_id": "main"
        },
        {
            "id": "ai_TXJ61ckM",
            "name": "lamp",
            "value": 0.9683093,
            "app_id": "main"
        },
        {
            "id": "ai_8LWlDfFD",
            "name": "table",
            "value": 0.96419907,
            "app_id": "main"
        },
        (...)
    ]
}
```

A.2.2. Detección de rostros

Herramientas evaluadas:

- Vision API - Face Detection
- Rekognition Images Features - Face Detection

Imagen utilizada para el análisis:

Figura A.2: Imagen seleccionada para el análisis de detección de rostros [25]

Respuestas obtenida por Vision API - Face Detection:


```
{
"faceAnnotations": [
    {
      "boundingPoly": {
        "vertices": [...]
      },
      "fdBoundingPoly": {
        "vertices": [...]
      },
      "landmarks": [
        {
          "type": "LEFT_EYE",
          "position": {
            "x": 937.46857,
            "y": 371.92477,
            "z": -0.000031891672
          }
        },
        {
          "type": "RIGHT_EYE",
          "position": {
            "x": 971.0181,
            "y": 395.2603,
            "z": 7.8464465
          }
```

```
        },
        {
          "type": "UPPER_LIP",
          "position": {
            "x": 927.3844,
            "y": 421.40594,
            "z": -3.7341352
          }
        },
        {
          "type": "LOWER_LIP",
          "position": {
            "x": 919.79004,
            "y": 431.8596,
            "z": 1.0571216
          }
        },
        {
          "type": "LEFT_EYE_PUPIL",
          "position": {
            "x": 933.96216,
            "y": 369.38736,
            "z": -1.6647142
          }
        },
        {
          "type": "RIGHT_EYE_PUPIL",
          "position": {
            "x": 973.3049,
            "y": 397.2248,
            "z": 7.2303495
          }
        },
        {
          "type": "LEFT_EYEBROW_UPPER_MIDPOINT",
          "position": {
            "x": 943.45624,
            "y": 355.63162,
            "z": -8.157934
          }
        },
        {
          "type": "RIGHT_EYEBROW_UPPER_MIDPOINT",
          "position": {
            "x": 985.1186,
            "y": 385.08423,
            "z": 1.2900349
          }
```

```
        },
      ],
      "rollAngle": 36.412548,
      "panAngle": 10.541368,
      "tiltAngle": -6.1371517,
      "detectionConfidence": 0.99150515,
      "landmarkingConfidence": 0.72887754,
      "joyLikelihood": "VERY_UNLIKELY",
      "sorrowLikelihood": "VERY_UNLIKELY",
      "angerLikelihood": "VERY_UNLIKELY",
      "surpriseLikelihood": "LIKELY",
      "underExposedLikelihood": "VERY_UNLIKELY",
      "blurredLikelihood": "VERY_UNLIKELY",
      "headwearLikelihood": "VERY_UNLIKELY"
    }
  ]
}
```


Respuestas obtenida por Rekognition Image Features - Face Detection:


```
{
    "FaceDetails": [
        {
            "BoundingBox": {...},
            "AgeRange": {
                "Low": 29,
                "High": 45
            },
            "Eyeglasses": {
                "Value": false,
                "Confidence": 99.98930358886719
            },
            "Sunglasses": {
                "Value": false,
                "Confidence": 99.53205108642578
            },
            "Gender": {
                "Value": "Male",
                "Confidence": 99.9288101196289
            },
            "Beard": {
                "Value": false,
                "Confidence": 85.27408599853516
            },
            "Emotions": [
                {
                    "Type": "HAPPY",
                    "Confidence": 12.762786865234375
                },
```

```
                {
                    "Type": "SAD",
                    "Confidence": 4.23891544342041
                },
                {
                    "Type": "CALM",
                    "Confidence": 2.8831183910369873
                }
            ],
            "Landmarks": [
                {
                    "Type": "eyeLeft",
                    "X": 0.6497119069099426,
                    "Y": 0.5892165303230286
                },
                {
                    "Type": "eyeRight",
                    "X": 0.6735051274299622,
                    "Y": 0.6256906986236572
                },
                {
                    "Type": "nose",
                    "X": 0.6542122960090637,
                    "Y": 0.6478393077850342
                },
                {
                    "Type": "mouthLeft",
                    "X": 0.6311430931091309,
                    "Y": 0.6539748311042786
                },
                {
                    "Type": "leftEyeLeft",
                    "X": 0.6453744173049927,
                    "Y": 0.5828641653060913
                },
                {
                    "Type": "leftEyeRight",
                    "X": 0.6541343927383423,
                    "Y": 0.5960025787353516
                },
                {
                    "Type": "mouthUp",
                    "X": 0.6434483528137207,
                    "Y": 0.6694350242614746
                },
                {
                    "Type": "mouthDown",
                    "X": 0.6395989060401917,
```

```
                    "Y": 0.6811977028846741
                }
            ],
            "Pose": {
                "Roll": 32.51116943359375,
                "Yaw": 14.94881534576416,
                "Pitch": 0.9145674705505371
            },
            "Quality": {
                "Brightness": 44.967498779296875,
                "Sharpness": 99.9305191040039
            },
            "Confidence": 99.98566436767578
        }
    ],
    "OrientationCorrection": "ROTATE_0"
}
```


A.2.3. Detección de textos

Herramientas evaluadas:

- Vision API - Text Detection
- Rekognition Images Features - Text Detection

Imagen utilizada:

Figura A.3: Imagen utilizada para la herramienta Vision API - Landmark Detection [32]

Respuestas obtenida por Vision API - Text Detection:


```
{
  "textAnnotations": [
    {
      "locale": "en",
      "description": "BUY ONE, GET ONE\nFREE\nWHOPPER\n"
    },
    {
      "description": "BUY"
    },
    {
      "description": "ONE"
    },
    {
      "description": ","
    },
    {
      "description": "GET"
    },
    {
      "description": "ONE"
    },
    {
      "description": "FREE"
    },
    {
      "description": "WHOPPER"
    }
  ]
}
```


Respuestas obtenida por Rekognition Image Features - Text Detection:


```
{
    "TextDetections": [
        {
            "Confidence": 99.80574035644531,
            "DetectedText": "BUY ONE, GET ONE",
            "Id": 0,
            "Type": "LINE"
        },
        {
            "Confidence": 99.81302642822266,
            "DetectedText": "BURGER",
            "Id": 1,
            "Type": "LINE"
        },
        {
```

```
            "Confidence": 97.22807312011719,
            "DetectedText": "KING FREE",
            "Id": 2,
            "Type": "LINE"
        },
        {
            "Confidence": 99.9590835571289,
            "DetectedText": "WHOPPER",
            "Id": 3,
            "Type": "LINE"
        },
        {
            "Confidence": 99.88521575927734,
            "DetectedText": "BUY",
            "Id": 4,
            "ParentId": 0,
            "Type": "WORD"
        },
        {
            "Confidence": 99.63867950439453,
            "DetectedText": "ONE,",
            "Id": 5,
            "ParentId": 0,
            "Type": "WORD"
        },
        {
            "Confidence": 99.80314636230469,
            "DetectedText": "GET",
            "Id": 6,
            "ParentId": 0,
            "Type": "WORD"
        },
        {
            "Confidence": 99.89590454101562,
            "DetectedText": "ONE",
            "Id": 7,
            "ParentId": 0,
            "Type": "WORD"
        },
        {
            "Confidence": 99.80762481689453,
            "DetectedText": "FREE",
            "Id": 10,
            "ParentId": 2,
            "Type": "WORD"
        },
        {
            "Confidence": 99.81302642822266,
```

```
            "DetectedText": "BURGER",
            "Id": 8,
            "ParentId": 1,
            "Type": "WORD"
        },
        {
            "Confidence": 94.64852905273438,
            "DetectedText": "KING",
            "Id": 9,
            "ParentId": 2,
            "Type": "WORD"
        },
        {
            "Confidence": 99.9590835571289,
            "DetectedText": "WHOPPER",
            "Id": 11,
            "ParentId": 3,
            "Type": "WORD"
        }
    ]
}
```


A.2.4. Detección de lugares de interés

Herramienta evaluada:

- Vision API - Landmark Detection

Imagen utilizada:

Figura A.4: Imagen utilizada para la herramienta Vision API - Landmark Detection. [27]

Respuesta obtenida:


```
{
"landmarkAnnotations": [
    {
      "mid": "/g/1jky8qnxv",
      "description": "Bethesda Fountain",
      "score": 0.37206264,
      "boundingPoly": {...},
      "locations": [
        {
          "latLng": {
            "latitude": 40.77417200000001,
            "longitude": -73.970968
          }
        }
      ]
    }
  ]
}
```


A.2.5. Detección de celebridades

Herramienta evaluada:

- Rekognition Image Features - Celebrity Detection

Imagen utilizada:

Figura A.5: Imagen utilizada para el análisis de las herramientas de detección de celebridades [29]

Respuesta obtenida por la herramienta:


```
{
"CelebrityFaces": [
        {
            "Urls": ["www.imdb.com/name/nm1101562"],
            "Name": "Simon Cowell",
            "Id": "6o4Oz",
            "MatchConfidence": 94
        },
        {
            "Urls": ["www.imdb.com/name/nm1123986"],
            "Name": "Louis Walsh",
            "Id": "30f17x",
            },
            "MatchConfidence": 90
        },
        {
            "Urls": ["www.imdb.com/name/nm0970122"],
            "Name": "Nicole Scherzinger",
            "Id": "3i6zp8c",
            },
            "MatchConfidence": 99
        },
        {
            "Urls": ["www.imdb.com/name/nm0651769"],
            "Name": "Sharon Osbourne",
            "Id": "m5y6R",
            },
            "MatchConfidence": 100
        }
    ],
    "UnrecognizedFaces": []
}
```


A.2.6. Detección de logos

Herramienta evaluada:

- Vision API - Logo Detection

Respuesta obtenida por la herramienta Vision API - Logo Detection para la imagen A.3.


```
{
 "logoAnnotations": [
    {
      "mid": "/m/01601q",
      "description": "Burger King",
```

```
      "score": 0.87624108,
    }
  ]
}
```


A.2.7. Detección de contenido inapropiado

Herramientas evaluadas:

- Vision API - Safe Search
- Rekognition Image Features - Image Moderation

Imagen utilizada:

Figura A.6: Imagen utilizada para el análisis de las herramientas de Vision API - Safe Search [36]

Respuesta obtenida por la herramienta Vision API - Safe Search:


```
{
"safeSearchAnnotation": {
    "adult": "POSSIBLE",
    "spoof": "UNLIKELY",
    "medical": "VERY_UNLIKELY",
    "violence": "VERY_UNLIKELY",
    "racy": "LIKELY"
  }
}
```

Imagen utilizada:

Figura A.7: Imagen utilizada para el análisis de las herramientas de Rekognition Image Features - Image Moderation [35]

Respuesta obtenida por la herramienta Rekognition Image Features - Image Moderation:


```
{
    "ModerationLabels": [
        {
            "Confidence": 65.83806610107422,
            "Name": "Suggestive",
            "ParentName": ""
        },
        {
            "Confidence": 65.83806610107422,
            "Name": "Female Swimwear Or Underwear",
            "ParentName": "Suggestive"
        }
    ]
}
```


A.3. Resultados análisis de herramientas de audio

A.3.1. Análisis de audio

Herramientas evaluadas:

- Cloud Speech API

Para esta herrmienta se utiliza el video “Steve Jobs Biography”. [59]

Respuesta obtenida por la herramienta Cloud Speech API:


```
{
    "results": {
          "alternatives": {
            "transcript":
            "Steve Jobs is an American entrepreneur and the co-founder of Apple
            Incorporated jobs was born in San Francisco in 1955 and its birth
            parents decided to give the child up for adoption the baby was adopted
            by Poland flower jobs and the young family move to California when Steve
            was 5 years old jobs achieve good grades at school and became friends
            with an older student named Steve Wozniak on the two friends began
            construction simple computers with all the students graduated in 1972
            and spend a short time at College before dropping out and attending
            a variety of creative classes for the next 18 months then took a position
            Atari in 1974 and what would Steve Wozniak on circuit board for video
            games and what\'s on various Technologies until they form their own
            company the Apple computer company",
    "confidence": 0.9663925766944885,
    "words": {
      "start_time": {
        "seconds": 13,
        "nanos": 400000000
      },
      "end_time": {
        "seconds": 13,
        "nanos": 900000000
      },
      "word": "Steve"
    },
    "words": {
      "start_time": {
        "seconds": 13,
        "nanos": 900000000
      },
      "end_time" {
        "seconds": 14,
        "nanos": 200000000
      },
      "word": "Jobs"
    },
    "words": {
      "start_time" {
        "seconds": 14,
        "nanos": 200000000
      },
      "end_time": {
        "seconds": 14,
        "nanos": 200000000
```

```
48          },
49          "word": "is"
50      }
51      (...)
52  }
```

Las pruebas mostradas anteriormente se pueden replicar en las siguientes direcciones:

- Vision API - https://cloud.google.com/vision/
- Cloud Video Intelligence - https://cloud.google.com/video-intelligence/
- Cloud Speech API - https://cloud.google.com/speech-to-text/
- Rekognition Image/Video Features[1] - https://console.aws.amazon.com/rekognition
- Predict Images/Video - https://clarifai.com/demo

[1]Para poder probar las funcionalidades de Amazon Rekognition se debe de crear una cuenta previamente.

Apéndice B

Motor de búsqueda para Google Custom Search

B.1. Motor de búsqueda para español

A continuación se listan las páginas utilizadas para formar el motor de búsqueda. Esto significa que se buscan los resultados que coincidan con las siguientes direcciones.

- *.twitter.com/*
- *.elpais.com/*
- *linkedin.com/*
- *.rae.es/*
- *.youtube.com/*
- *.facebook.com/*
- *.espndeportes.com/*
- *.wikipedia.org/*
- *.mercadolibre.com/*
- *.google.com/*

B.2. Motor de búsqueda para inglés

A continuación se listan las páginas utilizadas para Google Custom Search para videos en inglés.

- *.twitter.com/*

- *.nytimes.com/*
- *linkedin.com/*
- *.dictionary.cambridge.org/*
- *.youtube.com/*
- *.facebook.com/*
- *.foxsports.com/*
- *.wikipedia.org/*
- *.amazon.com/*
- *.google.com/*

Apéndice C

Comparación de índices

A continuación se muestran los índices generados en las distintas iteraciones para los primeros 20 segundos del video "FRIENDS - Season 5 Intro". [56]

C.1. Versión del índice en la iteración 2

Tag	**Start (s)**	**Duration (s)**
Black	0	0
Night	0	2
No Person	0	0
Human	2.24	39.9
People	2.24	39.9
Person	2.24	39.9
Adult	4	1
David Schwimmer	4.24	0.8
So	5.02	0
Has	5.07	0
Bed	5.07	0
Furniture	5.07	0
Matthew Perry	5.57	0.87
You	6.05	0.00
David Schwimmer	6.51	0.67
Flora	6.51	0.46
Jar	6.51	0.46
Plant	6.51	0.46
Potted Plant	6.51	0.46
Pottery	6.51	0.46
Vase	6.51	0.46

Tag	Start (s)	Duration (s)
Flora	8.17	0.81
Grand Piano	8.17	0
Jar	8.17	0.81
Leisure Activities	8.17	0
Music	8.17	0
Musical Instrument	8.17	0
Piano	8.17	0
Plant	8.17	0.81
Potted Plant	8.17	0.81
Pottery	8.17	0.81
Vase	8.17	0.81
Leo Legrand	8.31	0.13
Your	10.00	0
A	10.20	0
Asta	10.24	0
Blonde	10.24	0.94
Female	10.24	0.74
She	10.24	0.74
Girl	10.24	0.74
Her	10.24	0.74
Jennife Asta	10.24	0
Woman	10.24	0.74
Tecia Torres	11.31	0.13
Ennife	11.51	0
Jenni	11.51	0
Ston	11.51	0
Luggage	13.95	0
Suitcase	13.95	0
Is	15.05	0
Like	15.05	0
You're	15.10	0
Always	15.13	0
Luggage	15.55	0
Suitcase	15.55	0
Cdurteney Cox	16.68	0
Cox	16.68	0
Hair	16.68	0
Smoke	16.68	0

Tag	Start (s)	Duration (s)
Ten	16.68	0
Gevherhan Sultan	16.78	0
Couch	17.62	0.96
Furniture	17.62	0.96
Kudrow	17.62	0
Lsa Kudrow	17.62	0
Lisa Kudrow	18.72	0.17
Lisa	18.89	0
David Schwimmer	19.12	0.13
Hair	19.92	0
Lisa	19.92	0
Lisa Kudrow	19.92	0
When	19.92	0

Cuadro C.1: Índice generado en la segunda iteración para los primeros 20 segundos del video

C.2. Versión indice en la iteración 3

Tag	Start (s)	Duration (s)
Black	0	0
Night	0	2
No Person	0	0
Human	2.24	39.9
People	2.24	39.9
Person	2.24	39.9
Adult	4	1
David Schwimmer	4.24	0.8
Bed	5.07	0
Furniture	5.07	0
Matthew Perry	5.57	0.87
David Schwimmer	6.51	0.67
Flora	6.51	0.46
Jar	6.51	0.46
Plant	6.51	0.46
Potted Plant	6.51	0.46
Pottery	6.51	0.46
Vase	6.51	0.46

Tag	Start (s)	Duration (s)
Flora	8.17	0.81
Grand Piano	8.17	0
Jar	8.17	0.81
Leisure Activities	8.17	0
Music	8.17	0
Musical Instrument	8.17	0
Piano	8.17	0
Plant	8.17	0.81
Potted Plant	8.17	0.81
Pottery	8.17	0.81
Vase	8.17	0.81
Leo Legrand	8.31	0.13
Asta	10.24	0
Blonde	10.24	0.94
Female	10.24	0.74
Girl	10.24	0.74
Jennife Aniston	10.24	0
Woman	10.24	0.74
Tecia Torres	11.31	0.13
Ennife	11.51	0
Jenni	11.51	0
Ston	11.51	0
Luggage	13.95	0
Suitcase	13.95	0
Luggage	15.55	0
Suitcase	15.55	0
Courteney Cox	16.68	0
Cox	16.68	0
Hair	16.68	0
Smoke	16.68	0
Ten	16.68	0
Gevherhan Sultan	16.78	0
Couch	17.62	0.96
Furniture	17.62	0.96
Kudrow	17.62	0
Lisa Kudrow	17.62	0
Lisa Kudrow	18.72	0.17
Lisa	18.89	0

Tag	**Start (s)**	**Duration (s)**
David Schwimmer	19.12	0.13
Hair	19.92	0
Lisa	19.92	0
Lisa Kudrow	19.92	0

Cuadro C.2: Índice generado en la tercer iteración para los primeros 20 segundos del video

Apéndice D

Manual del programador

En esta sección se especifican los puntos a seguir para la configuración y ejecución el proyecto en un ambiente local. Asimismo, se introduce una guía de cómo añadir o quitar herramientas de análisis al proyecto.

D.1. Archivos importantes

Requirements.txt

En este archivo se encuentran todas las librerías de Python utilizadas.

Dockerfile

Archivo en el que se personaliza el contenedor que se utilizará para el despliegue de la aplicación.

App.yaml

En este archivo se configura el ambiente e infraestructura utilizada en Google App Engine.

D.2. Configuración de base de datos y herramientas de análisis

Para que el proyecto funcione de manera correcta, se deben configurar tanto la base de datos, como las herramientas de procesamiento utilizadas.

D.2.1. Configuración de la base de datos

Se debe crear una base de datos en Firebase. Para esto, se deben seguir los pasos especificados en `https://console.firebase.google.com/`.

Una vez creada, se deben indicar las credenciales generadas en el archivo: `/video_indexer/config/firebase_config.py`.

D.2.2. Configuración de herramientas de análisis

Para el uso de las herramientas de Google, Amazon y Clarifai, es necesario generar claves de autenticación. Para esto, se deben seguir los pasos indicados en las siguientes URLs:

- Google: `https://console.cloud.google.com/apis/credentials`
- Amazon: `https://aws.amazon.com/es/free`
- Clarifai: `https://clarifai.com/developer/account/signup`

Una vez obtenidas las credenciales, se deben agregar las mismas en el archivo: `/video_indexer/config/settings.py`

D.3. Configuración y ejecución del proyecto en ambiente local

A partir de este punto, se entiende que ya se realizaron los pasos mencionados en la sección anterior.

D.3.1. Configuración de ambiente

Python 3

Se debe contar con Python versión 3 instalado en el ordenador.

Instalación de librerías

Acceder a la carpeta raíz del proyecto, y ejecutar el comando: `sudo pip3 install -r requirements.txt`. Este instalará todas las librerías necesarias para la ejecución del proyecto.

D.3.2. Ejecución del proyecto

Una vez configurado el ambiente de forma correcta, para levantar el proyecto se debe acceder a su raíz y ejecutar el siguiente comando: `python3 main.py`.

D.4. Despliegue del proyecto en ambiente productivo

En la presente sección se detallan los pasos a seguir para hacer el despliegue de la aplicación de back-end en Google App Engine.

En primer lugar, se debe contar con un proyecto creado en Google Cloud Platform y un `DOCKER_ID` creado en Docker. Adicionalmente, es necesario instalar los SDK `gcloud` y `docker`. Para esto, se deben seguir los pasos especificados en las siguientes direcciones:

- Crear nuevo proyecto en Google Cloud Platform:

  `https://console.cloud.google.com/projectcreate`. Luego de esto, se le asigna al proyecto un identificador (`GOOGLE_CLOUD_PROYECT_ID`).

- Crear `DOCKER_ID`: `https://cloud.docker.com/`

- Instalar `gcloud`: `https://cloud.google.com/sdk/`

- Instalar `docker`: `https://docs.docker.com/get-started/`

Una vez realizados los pasos anteriores, ejecutar los siguientes comandos en la raíz del proyecto:

1. `sudo docker build -t gcr.io/GOOGLE_CLOUD_PROYECT_ID/NOMBRE_DE_VERSION`
2. `docker login`
3. Ingresar las credenciales asociadas al `DOCKER_ID` creado anteriormente.
4. `gcloud docker -- push gcr.io/GOOGLE_CLOUD_PROYECT_ID/NOMBRE_DE_VERSION`
5. `gcloud app deploy --image-url gcr.io/GOOGLE_CLOUD_PROYECT_ID/NOMBRE_DE_VERSION`

D.5. Pasos para añadir nueva herramienta de procesamiento

1. Se debe agregar el identificador de la nueva herramienta en el archivo

   `/video_indexer/config/settings.py`

2. Agregar el manejador del proveedor.

   - Si es una herramienta para análisis de audio:
     `/video_indexer/services/audio_clients`
   - Si es una herramienta para análisis de imágenes:
     `/video_indexer/services/images_clients`

- Si es una herramienta para análisis de videos:
  `/video_indexer/services/video_clients`

3. Modificar servicios correspondiente

   Para integrar el manejador creado en el punto anterior, debe ser utilizado por alguno de los siguientes servicios.

   - Si es una herramienta para análisis de audio se agrega en el servicio de audio:
     `/video_indexer/services/audio_service.py`
   - Si es una herramienta para análisis de audio se agrega en el servicio de imágenes:
     `/video_indexer/services/images_service.py`
   - Si es una herramienta para análisis de audio se agrega en el servicio de videos:
     `/video_indexer/services/video_service.py`

4. Por último, se debe indicar el umbral mínimo de confianza. Para esto, se debe agregar en la entidad *clients* de la base de datos, la información del umbral mínimo de confianza de la nueva herramienta.

Bibliografía

[1] 7 astonishing facts about youtube that you've never heard — huffpost. URL: https://www.huffingtonpost.com/kathleen-maloney/7-astonishing-facts-about{_}b{_}14630278.html. Último acceso: 10-06-2018.

[2] Amazon Rekognition: Object and scene detection. URL: https://console.aws.amazon.com/rekognition/home?region=us-east-1#/label-detection. Último acceso: 31-08-2018.

[3] Angular vs React: Comparación de 7 Principales Características. URL: https://code.tutsplus.com/es/articles/angular-vs-react-7-key-features-compared--cms-29044. Último acceso: 10-05-2018.

[4] Announcing Clarifai's new and improved v2 API video recognition feature — Clarifai. https://clarifai.com/blog/announcing-clarifais-new-and-improved-v2-api-video-recognition-feature. Último acceso: 28-04-2018.

[5] AWS - Capa gratuita de AWS. URL: https://aws.amazon.com/es/free/. Último acceso: 12-08-2018.

[6] Boto 3 Documentation. URL: https://boto3.readthedocs.io/en/latest/. Último acceso: 28-04-2018.

[7] Clarifai - demo. URL: https://www.clarifai.com/demo. Último acceso: 22-08-2018.

[8] Clarifai Python Documentation. URL: https://clarifai-python.readthedocs.io/en/latest/. Último acceso: 28-04-2018.

[9] Cloud Speech API. URL: https://cloud.google.com/speech-to-text/. Último acceso: 09-04-2018.

[10] Créditos de la nube de AWS para la investigación. URL: http://aws.amazon.com/research-credits. Último acceso: 02-09-2018.

[11] Diccionario de palabras en inglés y español. URL: http://corpus.rae.es/frec/CREA_total.zip. Último acceso: 12-06-2018.

[12] Dropping common terms: stop words. URL: https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html. Último acceso: 10-08-2018.

[13] English stopwords. URL: https://www.ranks.nl/stopwords. Último acceso: 12-03-2018.

[14] Ffmpeg. URL: https://www.ffmpeg.org/. Último acceso: 02-02-2018.

[15] Firebase. URL: https://firebase.google.com/?hl=es-419. Último acceso: 21-08-2018.

[16] Firebase Console. URL: https://console.firebase.google.com/. Último acceso: 12-08-2018.

[17] Firebase Realtime Database. URL: https://firebase.google.com/docs/database/. Último acceso: 21-08-2018.

[18] FLAC - comparison. URL: https://xiph.org/flac/comparison.html. Último acceso: 17-08-2018.

[19] Google-cloud Library Documentation. URL: https://googlecloudplatform.github.io/google-cloud-python/. Último acceso: 28-04-2018.

[20] Google Cloud Plataform - Credentials. URL: https://console.cloud.google.com/apis/credentials. Último acceso: 12-08-2018.

[21] Google Cloud Platform. URL: https://cloud.google.com/. Último acceso: 02-09-2018.

[22] Google Custom Search. URL: https://developers.google.com/custom-search/. Último acceso: 27-07-2018.

[23] Imagen de beyoncé knowles-carter. URL: https://s.hdnux.com/photos/25/21/22/5575832/3/920x920.jpg. Último acceso: 01-05-2018.

[24] Imagen de blake lively. URL: https://cd.cinescape.com.pe/cinescape-603x336-211387.jpg. Último acceso: 10-04-2018.

[25] Imagen de fernando muslera atajando. URL: https://img.fifa.com/image/upload/t_l4/nhnhj1te2yxix5qwvxio.jpg. Último acceso: 05-08-2018.

[26] Imagen de la fachada de m and m world en nueva york. URL: https://i.pinimg.com/originals/b6/0c/28/b60c28b83181ae63b24d033d1a85a918.jpg. Último acceso: 15-04-2018.

[27] Imagen de la fuente de bethesda en central park. URL: https://s3-eu-west-1.amazonaws.com/anuevayork/wp-content/uploads/2017/10/12112152/Que-ver-en-Nueva-York-en-un-dia-Central-Park-Bethesda-Fountain-800.jpg. Último acceso: 05-08-2018.

[28] Imagen de la guerra de afganistán. URL: https://static.todamateria.com.br/upload/gu/er/guerra-do-afeganistao-og.jpg. Último acceso: 19-04-2018.

[29] Imagen de los jueces de x factor (simon cowell, nicole scherzinger, sharon osbourne, louis walsh). URL: https://www.hellomagazine.com/imagenes/film/2016071532435/x-factor-trailer-2unlimited-no-limit/0-162-458/X-Factor-t.jpg. Último acceso: 05-08-2018.

[30] Imagen de luis suarez con la camiseta 9 del barcelona. URL: https://i.pinimg.com/originals/b6/0c/28/b60c28b83181ae63b24d033d1a85a918.jpg. Último acceso: 15-04-2018.

[31] Imagen de mujer haciendo yoga en la playa. URL: https://www.gettyimages.com/detail/photo/woman-doing-yoga-pose-on-beach-royalty-free-image/698703577. Último acceso: 20-04-2018.

[32] Imagen de un cartel de burger king. URL: https://c1.staticflickr.com/8/7025/6715723961_8e67bb34ec_b.jpg. Último acceso: 05-08-2018.

[33] Imagen de un dormitorio. URL: https://thumbor.gfrcdn.net//HdEQQBqzT0CAtOlEyMYLvncRpZg=/fit-in/320x320/https://s3.amazonaws.com/media.listamax.com/952504/1.jpg. Último acceso: 05-08-2018.

[34] Imagen de un hombre paseando un perro en el parque. URL: https://www.clearlylovedpets.com/blog/wp-content/uploads/2017/09/benefits-having-dog-1200x630.jpg. Último acceso: 12-04-2018.

[35] Imagen de una mujer corriendo. URL: https://www.phillymag.com/wp-content/uploads/sites/3/2016/03/runner-sprinting.jpg. Último acceso: 05-08-2018.

[36] Imagen de una persona cantando. URL: https://d39m9yulo0f19p.cloudfront.net/uploads/deals/216775/pictures/images_for_swf/767x320/4d08e950fcbb93cd8c78a08b0a45e827.jpg?r=f55fffb448f3dbff48af07d710d310fa. Último acceso: 05-08-2018.

[37] Imagen del logo de spotify. URL: http://www.scdn.co/i/_global/open-graph-default.png. Último acceso: 23-04-2018.

[38] Introduction to OpenCV. URL: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_setup/py_intro/py_intro.html#intro. Último acceso: 02-02-2018.

[39] Machine learning. URL: https://es.coursera.org/learn/machine-learning. Último acceso: 09-07-2018.

[40] Material design. URL: https://material.io/. Último acceso: 05-08-2018.

[41] Microsoft video indexer. URL: https://azure.microsoft.com/en-us/services/cognitive-services/video-indexer/. Último acceso: 23-08-2018.

[42] Modelos de datos. URL: https://clarifai.com/models. Último acceso: 04-04-2018.

[43] Precios de la herramienta cloud speech api. URL: https://cloud.google.com/speech-to-text/pricing. Último acceso: 01-05-2018.

[44] Precios de la herramienta cloud video intelligence. URL: https://cloud.google.com/video-intelligence/pricing. Último acceso: 01-05-2018.

[45] Precios de la herramienta vision api. URL: https://cloud.google.com/vision/pricing. Último acceso: 01-05-2018.

[46] Precios de las herramientas de rekognition. URL: https://aws.amazon.com/rekognition/pricing/. Último acceso: 23-04-2018.

[47] Pytube. URL: https://python-pytube.readthedocs.io/en/latest/. Último acceso: 02-02-2018.

[48] Quickstart for python in the app engine flexible environment. URL: https://cloud.google.com/appengine/docs/flexible/python/quickstart?hl=es. Último acceso: 05-08-2018.

[49] Requests: HTTP for humans. URL: http://docs.python-requests.org/en/master/. Último acceso: 10-05-2018.

[50] Resultado de búsqueda para "fsru2". URL: https://www.google.com.uy/search?rlz=1C5CHFA_enUY776UY776&ei=0IN4W6TYNsyEwgT-kr2ABw&q=fsru2&oq=fsru2&gs_l=psy-ab.3..0i19k1j0i13i30i19k1l2j0i8i13i30i19k1l7.179266.179266.0.179969.1.1.0.0.0.0.219.219.2-1.1.0....0...1c.2.64.psy-ab..0.1.219....0.Z-njWmMsW6A. Último acceso: 19-08-2018.

[51] Resultado de búsqueda para "Pen 1 - 1 Nac". URL: https://www.google.com.uy/search?q=Pen+1+-+1+Nac&rlz=1C5CHFA_enUY776UY776&oq=pen+1&aqs=chrome.0.69i59j69i57j0l4.1269j0j9&sourceid=chrome&ie=UTF-8. Último acceso: 19-08-2018.

[52] Spanish stopwords. URL: https://www.ranks.nl/stopwords/spanish. Último acceso: 12-03-2018.

[53] Top 6 reasons to use angular 5 framework for your project. URL: https://relevant.software/blog/top-6-reasons-to-use-angular-5-framework-for-your-project/. Último acceso: 10-05-2018.

[54] Urllib. URL: https://docs.python.org/2/library/urllib.html. Último acceso: 02-02-2018.

[55] Video: Coca Cola Life - First Kiss. URL: https://www.youtube.com/watch?v=DBMxqSJIPqs. Último acceso: 31-08-2018.

[56] Video: FRIENDS - Season 5 Intro B [HD]. URL: https://www.youtube.com/watch?v=aSTLzCYYXX4. Último acceso: 31-08-2018.

[57] Video: Girls Like You - Maroon 5. URL: https://www.youtube.com/watch?v=aJOTlE1K90k. Último acceso: 03-09-2018.

[58] Video One Guy, 43 Voices. URL: https://www.youtube.com/watch?v=jPoLeJJsbCw. Último acceso: 12-05-2018.

[59] Video: Steve Jobs Biography. URL: https://www.youtube.com/watch?v=mctZYeiNq9o. Último acceso: 31-08-2018.

[60] Watson - demo. URL: https://www.ibm.com/watson/services/visual-recognition/demo/#demo. Último acceso: 23-08-2018.

[61] What we do. URL: https://tryolabs.com/what-we-do/. Último acceso: 09-07-2018.

[62] XML Processing Modules. URL: https://docs.python.org/3/library/xml.html. Último acceso: 22-02-2018.

[63] YouTube API. URL: https://developers.google.com/youtube/. Último acceso: 28-04-2018.

[64] Kristin Houser. Amazon Rekognition falsely matched 28 members of congress to mugshots. URL: https://futurism.com/amazon-rekognition-falsely-matched-congresspeople/. Último acceso: 15-08-2018.