



Aseguramiento de la Calidad Software IC-6831

Primer Avance 1: Futbolito

Profesor: M. Sc. Saúl Calderón Ramírez

Estudiantes:

Carlos M. Girón Alas - 2014113159

Julián J. Méndez Oconitrillo - 2014121700

Daniel A. Troyo Garro - 2014073396



3 de setiembre, 2016.

Índice

1	Documento de Requerimientos ISO/IEC/IEEE 29148-2011	1
1.1	Propósito del sistema	1
1.2	Alcance del sistema	1
1.3	Panorama general del sistema.	2
1.3.1	Contexto del sistema	2
1.3.2	Funciones del sistema	2
1.3.3	Características del usuario	3
1.4	Requerimientos funcionales	3
1.5	Requerimientos de usabilidad	5
1.5.1	Entendibilidad	6
1.5.2	Operabilidad	6
1.6	Requerimientos de eficiencia	6
1.7	Interfaces del sistema	7
1.8	Operaciones del sistema	7
1.9	Modos y estados del sistema	8
1.10	Características físicas	9
1.10.1	Requerimientos físicos	9
1.10.2	Requerimientos de adaptabilidad	10
1.11	Condiciones de ambiente	10
1.12	Seguridad del sistema	10
1.13	Administración de la información	11
1.14	Regulaciones y políticas	11
1.15	Sostenibilidad del ciclo de vida del sistema	11
1.16	Empaquetado, manejo, envío y transporte	12
1.17	Verificación	12
1.18	Dependencias y supuestos	12
2	Estándares de Codificación	13
2.1	Estándar de codificación de java: Google Java Style.	13
2.2	Estándar de codificación de Javascript.	13
3	Diagramas UML	14
3.1	Diagrama de clases	14
3.2	Diagrama de componentes	15
3.3	Diagrama de casos de uso	16
3.4	Diagrama de actividad	17
4	Actividades de aseguramiento de la calidad de software IEEE-730	18

1. Documento de Requerimientos ISO/IEC/IEEE 29148-2011

1.1. Propósito del sistema

El Club Boca Juniors tiene la necesidad de mejorar los sistemas de análisis de los juegos con el fin de maximizar su rendimiento en la liga. Para ello, consideran que un sistema capaz de analizar los videos históricos de los partidos que ha tenido el equipo y que extraiga datos útiles de estos videos, les daría una visión más acertada de los elementos en los que pueden mejorar, tales como la solidez de su formación, el rendimiento individual de cada jugador y la forma en que interactúan los jugadores dependiendo de la posición en la que se encuentran.

Todos estos datos son extraídos estadísticamente del análisis de los videos. Este sistema pretende automatizar los análisis previamente mencionados, obteniendo y mostrando de la manera más clara y asimilable posible, los datos útiles para la mejora del rendimiento del equipo.

1.2. Alcance del sistema

Futbolito es un sistema que busca satisfacer las necesidades del equipo técnico y administrativo del Boca Juniors en cuanto a la revisión y obtención de datos de sus juegos anteriores.

El análisis manual de videos es una tarea exhaustiva, repetitiva y tediosa cuando es una persona el que debe hacerlo. Una situación en el que una persona tenga que ver cuatro partidos de 90 minutos de duración y analizarlos consumiría demasiado tiempo y recursos, además de no contar con la misma exactitud siempre. En cambio, dando uso a los algoritmos de análisis de visión por computadora, utilizando métodos probabilísticos y estadísticos junto con manejo de videos de forma automatizada, así como la identificación de elementos específicos en los partidos, puede reducirse la cantidad de tiempo que se invierte en el análisis de estos datos de forma sustancial. Dichos análisis puntuales se describen a continuación:

- División de cada video en escenas: Mediante el segmentado y posterior unión de todo el video en frames unitarios, se puede detectar un corte o cambio de escena tras un análisis probabilístico del tono (*hue* al pasar el frame al modelo de color HSV) de cada frame. De esta manera, se puede separar un video de un partido en distintos videos donde cada una es una escena del video original, esto para una selección más precisa de momentos específicos de un partido.
- Obtención de la duración de cada escena: Necesidad ligada a la anterior respecto a la selección de escenas. La determinación de este valor se da al calcular el número de frames por segundo y obtener la variable del tiempo.
- Eliminación de escenas irrelevantes como tomas del público, anuncios publicitarios, entre otros. El propósito principal de esto es el de filtrar segmentos del video que no aporten valor agregado al análisis tales como tomas del público, las cuales no aportan datos útiles que permitan generar un informe de la consistencia de la formación del equipo o de demás factores requeridos.
- Análisis de la consistencia y solidez de la formación del equipo en la cancha, pasando de la línea defensiva, medio campo y línea ofensiva. Mediante técnicas de *computer vision*, se logrará la identificación de cada jugador en una toma y el análisis colectivo de cada toma, se podrá saber la posición del jugador en el campo de juego a lo largo de todo el partido y

realizar un análisis estadístico de la influencia que tiene en el resultado final del partido este tipo de variables.

- Análisis del rendimiento individual de cada jugador: Tal y como se especificó en el punto anterior, tras realizar análisis estadísticos y probabilísticos sobre los datos obtenidos del análisis de posición de cada jugador, se puede derivar el desempeño que tuvo el jugador durante el partido y maneras de poder optimizarlo.
- Análisis del posicionamiento de cada jugador y su relación con la formación general del resto del equipo: Mediante identificación de la posición de cada jugador en el campo de juego en el video se puede obtener un conjunto de datos que representen el movimiento del jugador a lo largo del campo de juego durante todo el video y así poder determinar variables determinantes para aproximar un rendimiento del jugador, tales como velocidad, número de veces que rompió la formación general, número de *Fuera de juego*, entre otras.
- Generación de informes con los datos obtenidos durante todo el funcionamiento del sistema: Los usuarios principales del sistema, los cuales son el cuerpo técnico y junta administrativa del club Boca Juniors. Esto requiere que los datos obtenidos sean presentados de una manera clara, directa y explícita, brindando al usuario la mayor capacidad de comprensión de los reportes obtenidos. Por esto, el sistema generará archivos conteniendo los informes en formato .csv, .xlsx, .docx o pdf, donde se muestre el rendimiento general del equipo, así como los resultados de la segmentación temporal del video.

La aplicación principal de este sistema es la del análisis automatizado de videos de partidos de fútbol, principalmente hecho para el club Boca Juniors. Los objetivos y necesidades específicas del cliente ya han sido definidas en la sección anterior como ítems. Mediante dichos requerimientos se espera satisfacer las necesidades del cliente respecto a la obtención de datos estadísticos del equipo Boca Juniors.

1.3. Panorama general del sistema.

1.3.1. Contexto del sistema

La aplicación tiene varias etapas. Como es importante y pertinente que el sistema sea fácilmente accesible para el usuario que lo requiera en cualquier momento, se seleccionó la interfaz en un ambiente web, de manera que se pueda acceder desde cualquier equipo con un explorador web. En cuanto al desarrollo de las partes internas o back-end del sistema, es un servidor en java que atiende consultas de los clientes web y las atiende pertinentemente.

Las funcionalidades generales que tiene la aplicación web son la carga de un video y la selección de un tipo de análisis sobre el video subido al servidor. Una vez analizados los datos debe mostrársele al usuario los resultados del análisis en una forma amigable que sea fácil de entender. En el caso de la segmentación temporal, debe notarse cuáles son los momentos que presentan uno de los cambios de escena y cuáles no. En cuanto a los elementos de la identificación de jugadores se deben presentar los datos estadísticos de rendimiento de los jugadores y de posiciones adecuadas de acuerdo a la formación y a su desplazamiento por la cancha. Demás variables que el cliente considere apropiadas serán definidas en iteraciones posteriores del proyecto.

1.3.2. Funciones del sistema

El sistema principal debe constar de 3 elementos básicos. El primero es una aplicación web previamente descrita en la que el usuario puede acceder a las funcionalidades del sistema. Este primer elemento debe permitirle al usuario visualizar el video subido, así como los videos devueltos por el servidor; cargar videos para segmentación temporal y enviado al servidor, cargado de archivo *ground truth*, el cual puede ser de distintos formatos definidos más adelante, poder seleccionar los puntos de corte de manera manual mediante una interfaz cómodo y dinámica y poder descargar

los informes generados así como los videos productos de la segmentación.

El segundo elemento es el servidor web que debe tener una disponibilidad activa alta de forma que en casi cualquier momento se pueda usar las funcionalidades del sistema. Este elemento funciona como punto intermedio entre la plataforma web y la biblioteca de análisis de videos, enviando los videos e informes de un punto a otro y sirviendo como plataforma de ejecución del programa principal JAVA que realiza la lectura, análisis y segmentación del video.

El tercer y último elemento de mayor relevancia es la biblioteca utilizada para los análisis que se hacen sobre los videos. Dicha biblioteca funciona en conjunto con las biblioteca preestablecidas de OpenCV, framework multiplataforma que permite implementar algoritmos de *computer vision* y análisis de videos de manera simple y eficaz, para que ocurran los análisis probabilísticos y estadísticos de histogramas para los frames del video que desprenden otros datos de suma importancia que se pueden muestrear y aplicar distintos métodos que encuentren relaciones entre elementos. De momento la aplicación se limitará al análisis de videos de juegos de fútbol y la generación de datos estadísticos de los mismos, de forma que el uso que se le da a estos datos está limitado a lo que los usuarios decidan hacer con ellos. Esto implica que la subida de videos en los que no se presente un juego de fútbol no extraerá datos significativos al no poder identificar a los jugadores en el campo de juego y no obtendrá un análisis estadístico, mostrándosele un mensaje de advertencia.

1.3.3. Características del usuario

Los usuarios hacia los que está dirigida esta aplicación son el equipo técnico y administrativo del Boca Juniors. Sin embargo, la funcionalidad del sistema de análisis de videos podría tener aplicaciones para otros usuarios que den uso a la interpretación de señales. Este tipo de usuario final es muy amplio ya que es todo aquel que comprenda la aplicación que tiene los algoritmos de *computer vision* en el procesamiento de videos de partidos de fútbol y que tenga acceso a una terminal con un navegador web, por lo que un futuro el sector meta podría aumentar a otros clubes de fútbol o a libre acceso. En cuanto al equipo de operación, los elementos internos del servidor son manejados por una pareja de administradores o moderadores que se encargarán de casos anormales que el sistema mismo no es capaz de manejar. Por otra parte, en cuanto al mantenimiento de la aplicación, el equipo de mantenimiento consiste en dos técnicos que se encargan de la recolección de datos relevantes respecto al uso del sistema, y que están a cargo de posibles proyectos de integración, así como el modelado de nuevas funcionalidades y la integración de esta aplicación con otras que tengan elementos análogos que pueden aprovecharse.

1.4. Requerimientos funcionales

Los requerimientos funcionales se definen a continuación.

1.4.0.1 Acceso a la aplicación

En primer instancia el sistema debe tener una interfaz de usuario disponible en línea y esta debe contar como **mínimo** con la opción de subir el video y una vez que un video es seleccionado, la posibilidad de reemplazarlo o de enviarlo a análisis. Se le debe restringir al usuario la selección de archivos multimedia únicamente, siendo los siguientes formatos los inicialmente permitidos: .mp4, .avi, y .wav.

1.4.0.2 Subida del video

Compuesta de las siguientes etapas:

Opción de subir o reemplazar: La interfaz debe tener un botón claramente identificable que le permita subir un video al servidor sin la necesidad de hacer ningún estilo de comprobación manual por parte del usuario. .

Tiempo de espera: Durante el tiempo de subida del video al servidor, cuya duración es directamente proporcional al tamaño del video en cuestión, debe mostrársele al usuario una barra de avance que le informe cuánto tiempo estimado falta para que concluya la subida y cuánto tiempo ha transcurrido.

Subida misma del video: El sistema debe contar con el sistema de verificación que evita que el usuario suba un archivo que no sea multimedia con formato de video. En las características actuales, el sistema tiene soporte a .mp4, .wav y .avi, con búsquedas a añadir otros formatos.

Visualización del video: Tras subir el video a la aplicación web, esta debe mostrar un reproductor de video en el que el usuario pueda visualizar el video. Asegurándose de que la integridad del video no se vio comprometida en la subida del mismo y que cumpla con las características deseadas para el ingreso del mismo al componente del servidor que analizará el video.

Errores: El sistema debe mostrar al usuario un mensaje informativo sobre la naturaleza del error presentado, de una forma agradable. En el caso por ejemplo de un video subido con un formato no soportado, muestra el error. En caso de un problema en la subida del video, debe reportarse igualmente que el sistema encontró una dificultad en la subida que puede tener diversas naturalezas, en caso de que el sistema no esté disponible debe presentarse dicha información claramente y negar el uso del sitio durante el periodo de mantenimiento.

1.4.0.3 Envío del video a análisis

Compuesta de las siguientes etapas:

Opción analizar: La interfaz tiene un botón que permite al usuario enviar el video subido a análisis. Este botón debe ser habilitado cuando hay un video actual subido para ser analizado o al menos debe presentar un error cuando un archivo no ha sido subido.

Selección de punto de corte manual: Una vez cargado el video, se le debe permitir al usuario seleccionar uno o varios puntos entre los frames mostrados en pantalla (los cuales son obtenidos tras dividir el video por el programa java del servidor) y así el análisis entre videos se omitiría, ya que las escenas serían extraídas en base a lo seleccionado por el usuario. La selección de al menos un punto de división entre escenas anulará la segmentación temporal automática y el video se dividirá únicamente en las escenas determinadas por el usuario.

Tiempo de espera: Una vez presionado el botón de enviar a análisis, el sistema debe presentar de nuevo un estimado del tiempo que tomará el análisis del video para que el usuario tenga una idea del tiempo transcurrido y del posible tiempo restante.

Errores: En caso de que el análisis del video presentara algún error, debe presentarse un mensaje de error informativo al usuario. En caso de que el proceso se cierre o se actualice la página, debe redirigirse a la página de inicio por defecto, donde el usuario puede volver a comenzar el proceso de análisis sin la necesidad de volver a subir el video, o puede elegir reemplazar el video actual con un nuevo video en el botón de Subir video descrito en el requerimiento funcional 3.4.2.

1.4.0.4 Servidor web

El servicio web en java debe estar disponible en el momento que la página es utilizada. Esto implica que en caso de un error en el servidor, este debe informarse en la página web. El servicio cuenta con sistemas de respuesta únicamente a requests identificados de la página, por lo que la conexión se asegura como segura.

1.4.0.5 Algoritmos de segmentación temporal

Los resultados que presenta el sistema una vez analizado el video provienen de un análisis de histogramas obtenidos de cada cuadro o *frame* en los que es dividido el video que se subió para ser analizado. Todos estos algoritmos son implementados por el programa de java ejecutándose en el

servidor con la ayuda de funciones propias de la biblioteca *OpenCV*. De esta forma, se presentan a continuación dichas funciones.

Algoritmo de Bhattacharyya

El algoritmo de Bhattacharyya consiste en la obtención de la medida de distancia entre dos histogramas mediante el cálculo de un coeficiente homónimo y una posterior normalización utilizando un coeficiente obtenido con valores de los histogramas, tales como la media. La implementación de este algoritmo se realizará para calcular el grado de *diferencia* entre dos imágenes y así determinar si ocurre un *corte* o cambio de escena entre ellas, así poder separar el video en dos escenas en tal punto.

Mediante este algoritmo se calcula un arreglo de disimilitud entre cuadros del video consecutivos, almacenando así los valores obtenidos mediante el algoritmo de la distancia de Bhattacharyya.

Teorema de los tres sigmas

El teorema de los tres sigmas tiene una aplicación directa en el proceso de segmentación temporal y la obtención de los puntos de corte o de división de escenas. Este teorema es necesario para obtener en qué puntos del video ocurre un cambio de escena y así permite que mediante algoritmos de bibliotecas multimedia nativas del API de JAVA se logren la generación de nuevos videos, escenas que serán enviadas al usuario a la aplicación web.

1.4.0.6 Obtención de resultados

Una vez completado el proceso de análisis, el sistema cambiará su configuración para presentar los resultados obtenidos. Ahora, en lugar de presentarse los elementos iniciales, se presentarán un botón de avance y un botón de retroceso que permiten al usuario avanzar entre los frames procesados, y se muestra una etiqueta que le indica al usuario si en ese frame se presentó un corte. Se considera un corte a aquel frame justo después de uno con una escena completamente diferente. Compuesta de las siguientes etapas:

Obtención de escenas ordenadas por posición en el video inicial: La interfaz de la aplicación web debe mostrarle al usuario los N videos obtenidos de la segmentación del video con una etiqueta que indique su posición en el video inicial y la capacidad de poder visualizarlos en la aplicación, así como descargar un archivo comprimido que contenga todos estos videos. El archivo podrá ser de formato .zip.

Visualización de cortes detectados En algún punto de la aplicación web se le debe mostrar al usuario el número de cortes realizados al video original. Este es un valor simple, el cual está compuesto de un simple valor entero positivo.

Visualización de escenas Al igual como sucede con el video subido, la aplicación web tiene que permitirle al usuario visualizar cada una de los videos de las escenas devueltas tras la segmentación.

Muestreo de información obtenida del análisis: El sistema debe de mostrar la información en la página web obtenida del análisis del video. Estos valores pueden mostrarse de manera tabulada o mediante algún formato adecuado.

Descarga del informe en distintos formatos: El sistema debe de permitir la descarga del informe en formatos variados tales como .xlsx, .docx, .pdf o .csv.

1.5. Requerimientos de usabilidad

Los requerimientos de usabilidad respecto a la experiencia del usuario se describen a continuación.

1.5.1. Entendibilidad

El usuario debe comprender fácilmente lo que ocurre después de cada acción que realiza en el sistema así como el flujo general de las acciones del mismo. Se asegurará mediante una métrica de **completitud de la descripción** que el usuario logre dar un puntaje positivo a la facilidad de la comprensión del funcionamiento general del sistema. Esta encuesta será realizada mediante la herramienta Google Forms y sus valores serán obtenidos en la versión beta de la plataforma. Algunos componentes que influenciarán la entendibilidad del sistema son:

1.5.1.1 Mensajes de éxito, advertencias y errores

El sistema debe de mostrar de manera apropiada y entendible mensajes apropiados al comportamiento y reacción del sistema ante una entrada por parte del usuario. Los mensajes que debe de mostrar de la manera más clara posible son:

Error

Si se presenta un error, debe mostrársele un mensaje con los detalles y las opciones que tiene a continuación.

Advertencia

Si la acción que está por realizar tiene consecuencias no tan evidentes, un mensaje de advertencia deberá aclarar los que esto implica sin volverse un distractor ni obstruir el funcionamiento del sistema.

Éxito

Si un proceso ha concluido con éxito, el usuario debe ver los elementos que se completaron y las opciones siguientes para continuar en el uso de la página si así lo desea.

1.5.2. Operabilidad

El sistema debe de permitir al usuario interactuar con él de una manera eficaz y que ningún dato de entrada inesperado colapse el funcionamiento. Por esto, se utilizará la métrica de **chequeo de la validez de los datos de entrada**, la cual se realizará en etapas de desarrollo desde el programa del servidor desarrollado y utilizando la herramienta de JUnit, realizando distintas pruebas de aceptación.

1.6. Requerimientos de eficiencia

En cuanto a la eficiencia, el sistema debe garantizar los siguientes elementos.

1.6.0.1 En cuanto a la carga y subida de los elementos

El sistema maneja cargas y subidas de videos y de imágenes. El tiempo de respuesta del sistema es absoluto y creciente de acuerdo al tamaño de los archivos que se manejan. Se utilizará como métricas al **tiempo de respuesta** y el **consumo de recursos** para garantizar la calidad de la eficiencia del sistema procesando los videos.

1.6.0.2 En cuanto a la disponibilidad espacial

El sistema debe ser accesible desde cualquier navegador web Google Chrome, Mozilla Firefox u Opera con una conexión activa a internet.

1.6.0.3 En cuanto a la disponibilidad de procesamiento

Un equipo que tenga la capacidad de correr un navegador actual y que este soporte JavaScript tiene los requerimientos suficientes para correr esta aplicación. La carga principal del procesamiento del video será dejada al servidor, quien realizará la conexión con la base de datos de Boca Juniors y de demás operaciones, permitiendo que la terminal cliente no consuma recursos considerables.

1.6.0.4 Tiempo de vida mínimo esperado

Se estima un tiempo de vida estable de la aplicación durante entre 2 a 5 años.

1.6.0.5 La duración esperada de una sesión

Una única sesión puede tener una duración indefinida y sin interrupciones de aproximadamente 1 día completo. En una sesión se actualiza únicamente una variable con datos del usuario y todo lo demás ocurre en el servidor.

1.7. Interfaces del sistema

La interfaz de usuario está en una página web que debe ser fácilmente accesible y entendible para el usuario que la utiliza.

1.7.0.1 Navegación del sitio

Se compone de los siguientes elementos

Flujo natural de la página

El sistema debe avanzar en un orden lógico entre la página de inicio y la página de resultados. Esta última debe ser accesible únicamente cuando se han cumplido los requerimientos para que esta puede accederse. Esto significa que bajo ninguna circunstancia debe mostrarse una página de resultados sin que haya un video en la sesión actual.

Retroceso

En caso de que el usuario quiera retroceder una vez alcanzada la pantalla de resultados, la interfaz debe tener un botón que le permita volver a la página original donde puede seleccionar un video diferente. Esta funcionalidad es equivalente a recargar la página.

1.7.0.2 Disposición espacial de los elementos en la interfaz

El sistema debe ser fácilmente comprensible para un usuario que nunca antes ha utilizado la aplicación pero que sabe para qué es. En caso de que el usuario no sepa en qué consiste la aplicación que se está desarrollando, al menos es importante que el usuario tenga un acceso rápido a la información del uso de la página.

El usuario además debe poder identificar rápidamente y sin distractores innecesarios las opciones que tiene disponibles y qué significan. Esto es, que el menú en el que se encuentran las opciones para el usuario esté en un lugar visible y sea fácilmente distinguible del resto de la interfaz, así como los íconos de ayuda y acceso a la documentación.

1.8. Operaciones del sistema

Se subdividen en los siguientes elementos.

1.8.0.1 Requerimientos de integración del sistema humano

La tarea que se desarrolla en esta aplicación no tiene ningún nivel relevante de criticidad. Esto significa que la tarea no debe tener un cuidado especial con los usuarios que lo utilizan, y el valor agregado que tiene es únicamente aplicable para los usuarios que buscan datos estadísticos a partir de un video mediante el análisis estadístico que se implementa en esta aplicación.

1.8.0.2 Mantenibilidad

Los requerimientos específicos de mantenibilidad se cuantificarán de la siguiente forma.

Tiempo

El tiempo de respuesta para una falla técnica que un usuario reporte es de 2 días. En caso de ser una falla grande en el servidor se estiman tiempos de respuesta de menos de una semana pues es el tiempo que toma establecer un servidor funcional con la capacidad de reemplazar completamente el anterior con una versión nueva 100 % funcional. En esta aplicación no hay persistencia de datos y es por eso que el sistema no tiene la necesidad de guardar los videos ni de recuperar ninguna información que no sea persistente.

Ritmo

Se plantea un equipo de mantenimiento capaz de resolver los problemas al tiempo que aparecen, y que hace un chequeo general de mantenimiento semanal.

Complejidad del Mantenimiento.

Como ya se explicó en la sección de Tiempo, una reconstrucción completa del sistema base puede lograrse en una semana por lo que se considera que la complejidad del mantenimiento es baja. El diseño de clases y las relaciones entre los elementos cumple los patrones básicos de diseño que facilitan el mantenimiento lo más posible.

Índices de acción de mantenimiento

Cada hora se estima en \$50 por hora utilizada en labores de mantenimiento, para cada miembro del equipo de mantenimiento.

1.8.0.3 Confiabilidad

El equipo de mantenimiento está disponible 40 horas semanales y pretende dar soporte constante en un tiempo menor a una semana. El sistema por otra parte, salvo elementos sin precedente o inesperados como problemas espaciales, el sistema está disponible 24 hrs al día.

1.9. Modos y estados del sistema

Al momento de la elaboración de este documento, el sistema presentará tres estados o modos expuestos al usuario, los cuales serán los siguientes:

- **Estado sin sesión:** En este estado la aplicación web se mantiene en la pantalla de inicio de sesión, aguardando que el usuario ingrese los datos para iniciar su sesión y que puede operar la plataforma. En este modo no se permiten acciones adicionales al ingreso de los datos de seguridad para el inicio de sesión.

- **Manejo de cuenta y de la plataforma:** En este estado se le permite al usuario ver los informes generados de anteriores videos cargados, así como demás información que sea considerada importante como la fecha de subida y el usuario que subió el video. Además se le permite al usuario el cambio de datos de su cuenta tales como el nombre de usuario o la contraseña.
- **Modo de espera de carga del video:** En este estado el sistema se encuentra a la espera de que el usuario cargue a la página un video para así poder posteriormente procesarlo y segmentarlo. No se muestra ninguna información de resultados relevante y es el modo más importante del sistema al ser el necesario para el ingreso de información y datos. El sistema mostrará mensajes de error si se trata de subir un video no válido (formato no aceptado o errores similares) o un archivo corrupto.
- **Modo de selección de cortes y visualización de video subido:** El estado en el que entra el sistema tras de que se detecte y cargue el video inicial. En este modo el usuario puede visualizar el video subido, subir un archivo *ground truth* con la información de los cortes o escenas deseadas y además se le permite al usuario seleccionar los puntos entre frames del video donde se realizará un corte de escena. El usuario podrá pasar aquí al modo de Estado de procesamiento al enviar el video.
- **Estado de procesamiento:** Estado en el que se muestra el porcentaje carga del sistema respecto a la carga del video o al análisis total del video, así como la generación de informes. En este estado únicamente se le muestra al usuario el valor de completitud de la tarea que actualmente esté realizando el sistema.
- **Muestreo de resultados:** Tras realizar un análisis exitoso de un video subido al sistema, este devolverá una serie de resultados tales como el número de escenas obtenidas, el momento en el video en el que termina y comienza cada escena, la visualización de cada escena y demás datos relacionados al rendimiento del equipo y de cada jugador a nivel individual. Se le permitirá al usuario archivar estos resultados para que sean consultados después por él o demás o miembros de la plataforma. También se le permitirá descargar informes generales de los resultados obtenidos del video en distintos formatos. Es el estado que más información brinda de los datos pertenecientes al sistema, junto con *Manejo de cuenta y de la plataforma*.

A continuación, se puede visualizar un diagrama de estados indicando el comportamiento del sistema y transiciones de los diagramas anteriormente descritos.

1.10. Características físicas

Las características físicas del producto se describen a continuación.

1.10.1. Requerimientos físicos

El sistema debe estar hospedado en una ubicación con acceso a internet las 24 hrs al día de forma que el servidor se conecte de forma indiferente a los clientes que entren a usar el sistema en todo momento. Como el uso de la aplicación es limitado a pocos usuarios inicialmente y luego a pocos usuarios que podrían darle uso distinto a la aplicación, la disponibilidad no debe ser absoluta y no debe ser reemplazable en caso de fallas en un tiempo menor al especificado en la sección 3.8.2. El sistema estará hosteado en un servidor de Amazon Web Services.

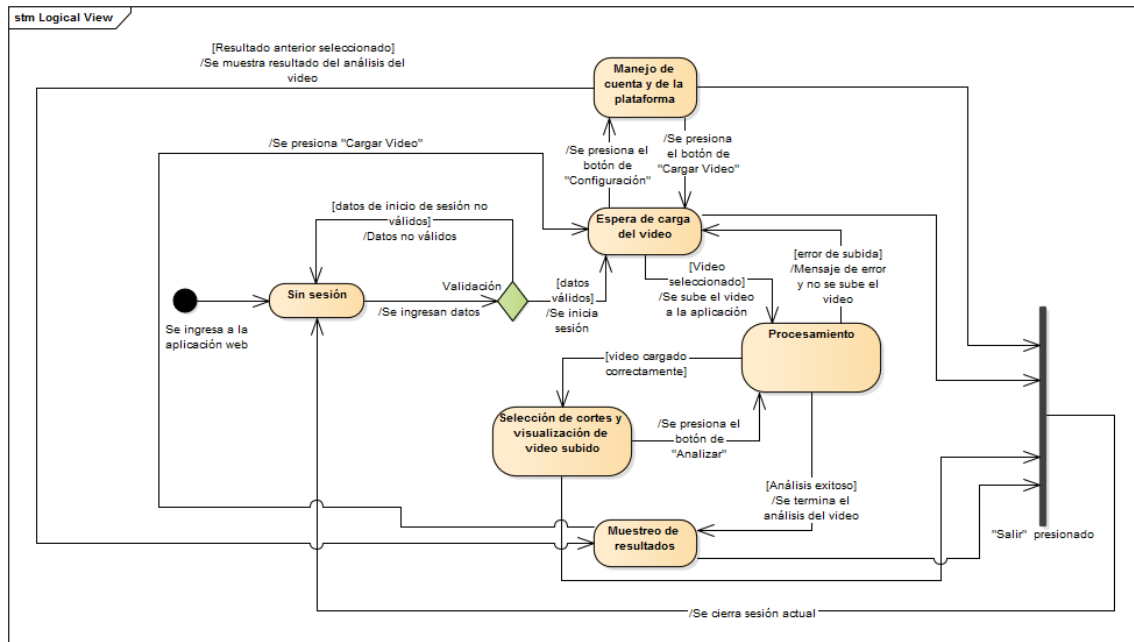


Figura 1. Diagrama de estados

1.10.2. Requerimientos de adaptabilidad

El sistema se estima que en un futuro podría integrarse como un módulo de un sistema de trabajo de videos para encontrar patrones o analisis extensivo de información en grandes cantidades. Así, se considera que deben ser completamente intercambiables las funciones que analizan los archivos de video así como aquellas que generan las estadísticas y datos de valor respecto al contenido analizado.

1.11. Condiciones de ambiente

Debido a que el servicio de hosting será brindado mediante Amazon Web Services. Por este motivo, se exime de toda responsabilidad respecto a las condiciones ambientales del servidor al equipo original de desarrollo y se le delegan a la compañía externa mediante la obtención de sus servicios.

1.12. Seguridad del sistema

El sistema debe de validar el acceso de cada usuario mediante un sistema de *Log In*, el cual tendrá una validación única mediante el nombre de usuario y una contraseña asociada a dicha cuenta. Esto se debe a que el sistema almacenará datos importantes acerca del rendimiento histórico de los jugadores en los partidos anteriores, lo cual podría ser considerado como información valiosa que debe mantenerse confidencial.

El almacenaje de las cuentas se dará por parte del departamento de TI (*Tecnologías de Información*) con una base de datos que ellos poseen y administran, por lo que la seguridad de estas cuentas. Esto se explicará más adelante en la sección 3.18 **Dependencias y Supuestos**. Adicional a este requerimiento de seguridad, ningún otro requerimiento adicional fue solicitado por el cliente. Se exime de la integridad física del edificio en el que se encuentra el servidor del hosting como de la base de datos al equipo de desarrollo, ya que, como se explicó en el apartado anterior, son responsabilidades delegadas a la compañía externa.

1.13. Administración de la información

El sistema recibirá únicamente archivos de video, entre los que se permitirán archivos de extensión .mp4, avi, wmv, entre otros. Adicional a este tipo de archivos, se le permitirá también el registro de archivos .txt, .xml o .json como forma de archivos *ground truth* para la segmentación temporal manual del video. Todos estos tipos de archivos pueden ser fácilmente ingresados al sistema mediante la aplicación web.

El componente del servidor, que devolverá directamente archivos de video en formato .avi, los cuales vendrán a representar las escenas del video, también podrá devolver archivos contenedores de información. Estos últimos archivos contendrán los informes generados y podrán ser generados como .csv, .xml, .json o .txt. La información atómica de estos informes será almacenada en la base de datos de forma más unitaria, pudiendo estar en distintas tablas a través de distintos esquemas. Durante el desarrollo del sistema se podrán añadir nuevos requerimientos que requieran manejo adicional de la información, por lo que esta sección puede actualizarse.

1.14. Regulaciones y políticas

La plataforma web estará disponible en dos idiomas, siendo estos el inglés y el español. Puesto que el contrato es para un club de fútbol argentino, se seguirán conceptos y regulaciones establecidas por la FIFA para la delimitación y segmentación del campo, así como demás métricas y restricciones establecidas en los reglamentos acordes.

Se requerirá que cada cuenta esté asociada a un miembro del cuerpo técnico del club como de la junta administrativa, por lo que dichos datos serán requeridos a la hora de ingresar una cuenta y validados por el equipo de soporte del sistema. Estos datos serán albergados en la base de datos de manera encriptada, por lo que permanecerán seguros a posibles filtraciones o robos de información. Adicional a estas regulaciones ninguna otra política es requerida por parte del equipo de desarrollo como del club Boca Juniors.

1.15. Sostenibilidad del ciclo de vida del sistema

Se establecerá la metodología ágil SCRUM para realizar un proceso de desarrollo de proyecto dinámico y ordenado, con el motivo de obtener la mejor calidad de software posible. En esta metodología se irán realizando sprints o iteraciones de desarrollo donde se trata de dejar en el mejor estado posible un componente entregable. Este componente será diseñado, programado y probado durante cada sprint. La metodología SCRUM dictamina la elaboración de reuniones al final de cada sprint llamadas Retrospectivas. Esto con el fin de extraer datos útiles de cada miembro del equipo y establecer o alterar objetivos y el rumbo general del proyecto.

También se asegurará que los desarrolladores reciban el proceso de capacitación apropiado para el manejo de las bibliotecas necesarias de análisis de video y *computer vision*. Esto implicará el pago de un curso en plataformas educativas tales como *edX* y *coursera*. Los desarrolladores tienen la opción de tomar estos cursos o no, pero se les impulsa a tomarlos para disminuir el tiempo de la curva de aprendizaje y obtener un resultado más eficiente.

Como actividad de aseguramiento de la calidad del software, se establecerán estándares de codificación para los lenguajes utilizados mayoritariamente, que en este proyecto son Java y Javascript. Los estándares escogidos fueron Google Java Style y el estándar determinado del verificador online jshint. Estos estándares facilitarán el mantenimiento de la herramienta y extenderá la vida del producto final.

También se realizará una verificación de los requerimientos, diseño conforme a los requerimientos y código respecto al diseño al final de cada iteración del desarrollo. Se le consultará al cliente al realizar un cambio en el diseño de alto nivel o cualquier duda respecto a los requerimientos, siempre al final de una iteración o sprint. Estos procesos serán acorde al objetivo establecido en el estándar IEEE-730, aunque no se seguirá este estándar al pie de letra.

1.16. Empaquetado, manejo, envío y transporte

Dado que el sistema se accede mediante una aplicación web y no es siquiera un bien físico, esta sección será completamente omitida. Se asume que el cliente final ya posee una terminal con un navegador web y acceso a Internet, tal y como se especifica más adelante en la sección 3.18.

1.17. Verificación

La verificación de los componentes se logrará mediante pruebas al final de cada iteración o etapa del desarrollo. Se recuerda que se utilizará SCRUM como metodología de trabajo y que esto requiere la realización de sprints iterativos. Tras cada sprint se utilizará la herramienta JUnit para la realización de test unitarios, en los componentes programados en JAVA, y de integración con los componentes web.

Las pruebas de aceptación serán realizadas emulando los casos de uso en el que podrá estar el usuario al utilizar el sistema en su totalidad. Así, un miembro del equipo de desarrollo emulará dicho caso y ejecutará todas las funcionalidades de la misma manera de la que lo haría un usuario promedio. Si todo el proceso es exitoso, se considera que la prueba fue validada.

1.18. Dependencias y supuestos

Para la correcta aplicación del sistema se asume que el usuario, grupo que como ya se ha mencionado varias veces, consistirá inicialmente de los miembros del cuerpo técnico del club Boca Juniors y de la Junta Administrativa de dicha agrupación; presenta un navegador de internet que soporte Javascript y una conexión a Internet mayor a 512Kb/s. Factores adicionales como el sistema operativo de la terminal del cliente no son importantes en este proyecto ya que se accede al sistema mediante la aplicación web.

A nivel del servidor se asume que los servicios brindados por AWS se integrarán sin problemas a los datos albergados en el hosting. Problemas con esta integración no serán responsabilidad del equipo de desarrollo al ser delegadas mediante *outsourcing*.

Respecto al almacenaje de las cuentas, esto será delegado al Club Boca Juniors mediante la Intranet que está bajo su propiedad. Esta base de datos almacena tablas con la información de los miembros pertenecientes al club y datos de la cuenta de estos miembros, así como los informes de los videos generados enlazados a cada usuario. Se supone que esta base de datos será actualizada y recibirá mantenimiento externo al del equipo de desarrollo. Mediante un componente de conexión, los datos de inicio de sesión serán obtenidos de esta base de datos. La seguridad así como la integridad de los datos se le delega al departamento de TI del Boca Juniors.

2. Estándares de Codificación

2.1. Estándar de codificación de java: Google Java Style.

Para mantener un mayor orden respecto al código de los componentes de Java se escoge el estándar de Google, o conocido como *Google Java Style*. Esta guía de formato del código permite una clara y mayor comprensión del código escritor por el programador y además permite mantener así un formato compartido por los miembros del equipo que programen en el lenguaje JAVA. Factores como la popularidad del estándar, herramientas de verificación de código disponibles, la empresa que respalda el estándar y el tiempo que el estándar ha sido utilizado fueron determinantes para escoger dicho estándar.

La herramienta seleccionada para la verificación del código del proyecto es **Checkstyle**, popular herramienta en Java que verifica que el código analizado se adhiera a un estándar de codificación determinado. Presenta compatibilidad y acoplabilidad con el ambiente de desarrollo Eclipse, por lo que es ideal para este proyecto.

2.2. Estándar de codificación de Javascript.

La página JSHint es una herramienta poderosa para el chequeo del estándar de código por defecto que asegura la calidad del software desarrollado en JavaScript. La dirección de la página es <http://jshint.com/>. La documentación de los casos específicos está en <http://jshint.com/docs/> y se especifican casos especiales que revisa comentarios que se pueden hacer para que quede más claro lo que se hizo.

Escogimos esta herramienta porque al ser web es verdaderamente sencilla respecto a las herramientas que requieren instalación, y maximiza la portabilidad, de forma que se puede programar elementos del proyecto en un equipo sin preparación previa ni herramienta instalada y revisar rápidamente que cumpla el estándar de calidad. Para hacer todavía más valor a esa portabilidad, usamos la configuración por defecto de la página, a pesar de que es posible cambiarla.

3. Diagramas UML

Adjunto al archivo comprimido donde se ubica este documento, puede encontrar las imágenes de los diagramas en una mejor calidad y mayor resolución.

3.1. Diagrama de clases

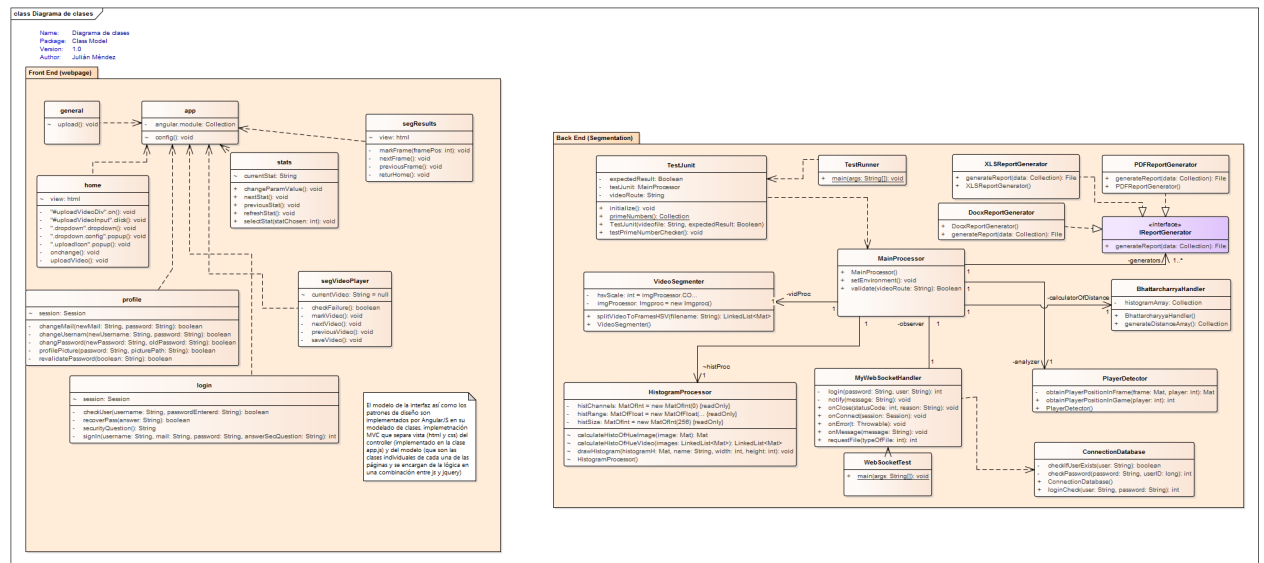


Figura 2. Primera iteración del diagrama de clases del sistema.

En la anterior imagen se puede apreciar al diagrama UML de clases. Este diagrama muestra dos secciones principales, las cuales agrupan al front-end y al back-end. Puesto que no hay una relación directa entre ellas, no se presentan relación alguna. Este diagrama es sólo una primera iteración y puede cambiar conforme avance el proyecto.

Un patrón de diseño presentado en el diagrama es el patrón de comportamiento **Observer**, el cual se puede visualizar entre las clases `WebsocketHandler` y `MainProcessor`, clase que maneja el flujo de control del algoritmo de segmentación temporal e identificación de jugadores. En este caso, una instancia de `WebsocketHandler` mantiene como observador a una instancia de clase `MainProcessor`, de modo tal que apenas llegue un mensaje de la aplicación web se iniciará el proceso de segmentación temporal, notificándole al observador que comience. La explicación de este patrón la puede encontrar en la sección de *Behavioral Design Patterns* del libro *Design Patterns: Elements of Reusable Object-Oriented Software*, indicado en la sección de Referencias.

3.2. Diagrama de componentes

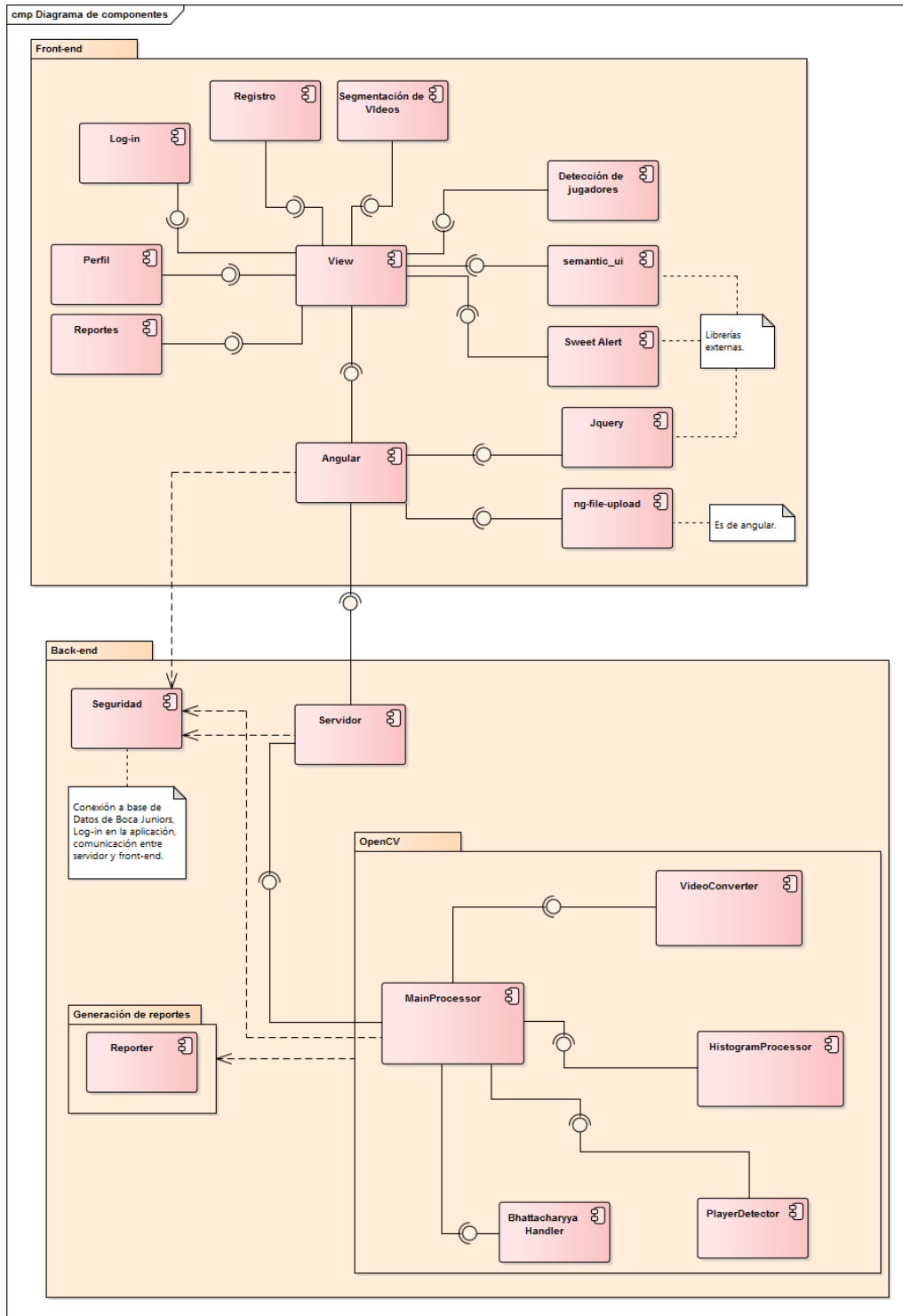


Figura 3. Primera iteración del diagrama de componentes del sistema.

El diagrama de componentes del sistema muestra las unidades de componentes que fueron inferidas en esta primera iteración del proyecto. Es muy similar al diagrama de clases, por lo que no se incurrirá a explicación más detallada. Componentes externos y que simplemente son utilizados mediante interfaces son denotados como una nota en el diagrama de componentes.

3.3. Diagrama de casos de uso

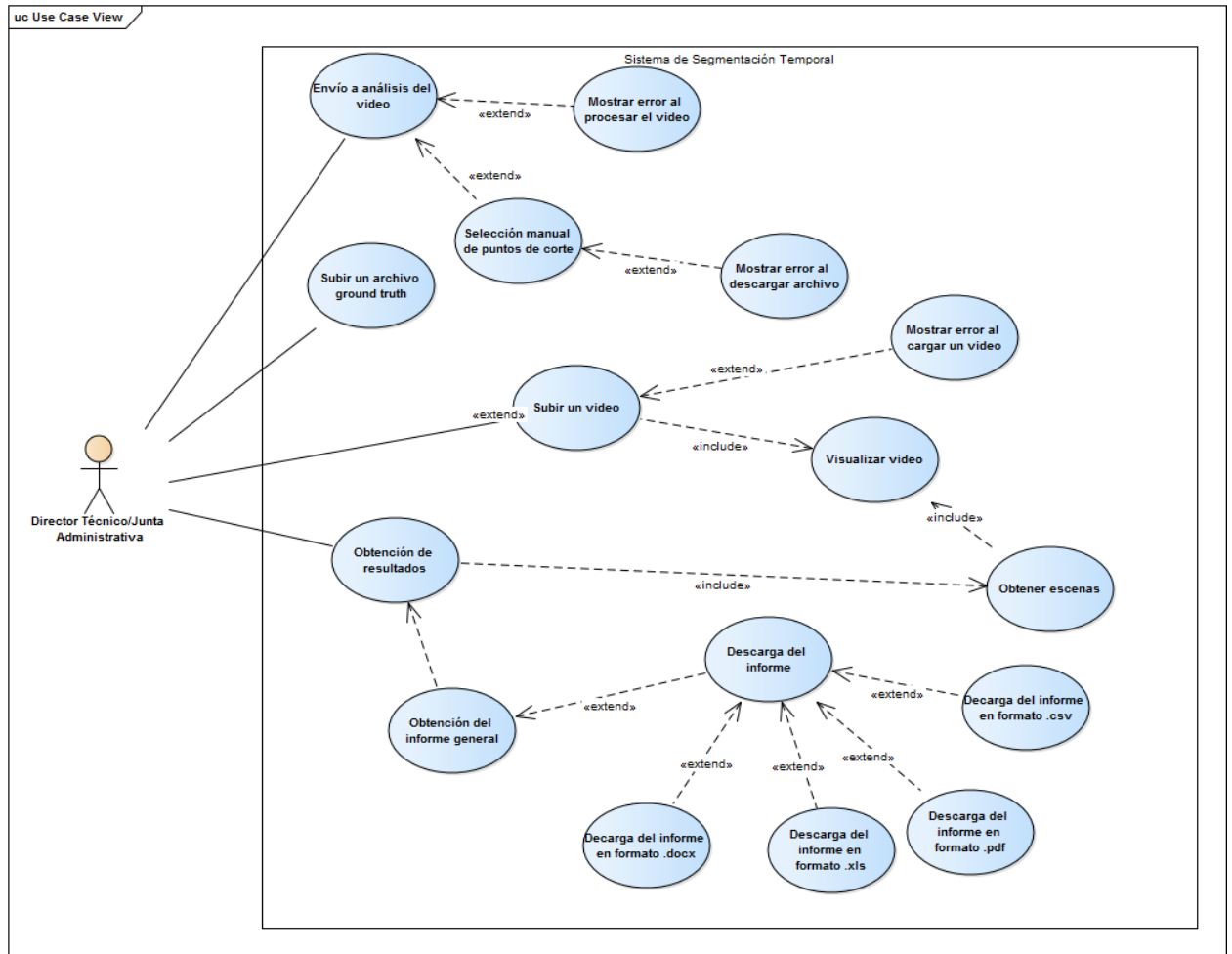


Figura 4. Primera iteración del diagrama de casos de uso.

El diagrama de casos de uso presenta las funcionalidades y posibles escenarios en los que el usuario principal de este sistema, que son los miembros del cuerpo técnico y junta administrativa del Club Boca Juniors, se expondrá. Es una versión no definitiva al ser la primera iteración del proyecto.

3.4. Diagrama de actividad

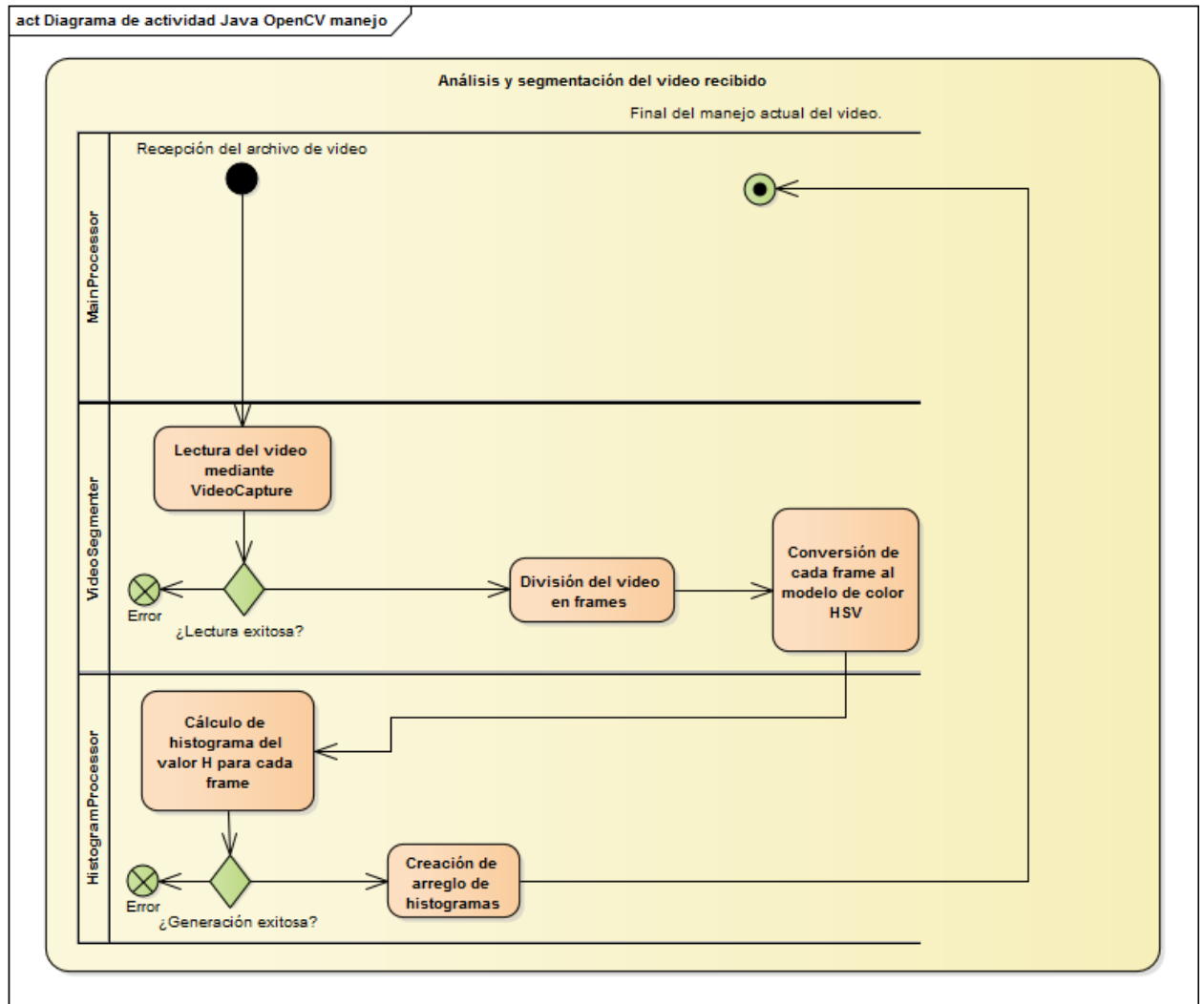


Figura 5. Diagrama de actividad incompleto del proceso de segmentación temporal del video. Análisis de Bhattacharyya pendiente.

En el presente diagrama de actividad se muestra el flujo inicial incompleto de los procesos que realiza el programa de JAVA ejecutándose en el servidor. Este diagrama se muestra incompleto ya que en esta iteración no se logró el completado de los cálculos de la distancia de Bhattacharyya.

4. Actividades de aseguramiento de la calidad de software IEEE-730

En esta sección se mostrarán las actividades que se realizarán a lo largo del proceso de desarrollo de software para asegurar valores de calidad según establece el estándar IEEE-730. A continuación se muestra la tabla con las actividades establecidas.

Actividades de Aseguramiento de la Calidad del Software			
Actividad	Responsable	Actividad concreta	Sprint asignado (si no se aplica al siguiente)
Planeamiento			
Definición de métricas para establecer un valor de calidad.	Carlos Girón, Daniel Troyo y Julián Méndez	-	Inicio del proyecto
Selección y configuración de una herramienta para administración de proyectos.	Daniel Troyo	-	Inicio del proyecto.
Asignación de actividades y responsables.	Daniel Troyo	-	Inicio del proyecto
Selección y aprendizaje de una metodología ágil de trabajo. .	Daniel Troyo	-	Inicio del proyecto
Proceso			
Evaluación de conformidad de los componentes web.	Carlos Girón y Julián Méndez	Revisión del código Javascript con la herramienta JSHint al final del sprint.	-
Evaluación de conformidad de los componentes java del servidor.	Carlos Girón y Julián Méndez	Revisión del código Java con la herramienta CheckStyle al final del sprint, para que cumpla el estándar Google Java Style.	-

Evaluar los valores de las métricas inicialmente definidas y establecer la diferencia con las deseadas inicialmente. Esto al final del sprint.	Daniel Troyo y Julián Méndez	Aplicar las herramientas obtenidas para cada métrica.	-
Análisis de la satisfacción del cliente respecto al avance actual del proyecto.	Carlos Girón, Daniel Troyo y Julián Méndez	Reunión con el cliente en donde se le muestra el avance actual y se evalúa su opinión respecto al producto.	-
Conocer al equipo de mantenimiento o IT del Boca Juniors .	Carlos Girón y Julián Méndez	Visita a las instalaciones de TI del Boca Juniors..	-
Evaluar la idoneidad del avance realizado.	Carlos Girón	Realización de pruebas en JUnit para medir la funcionalidad adecuada.	-
Producto			
Evaluación del seguimiento del código acorde al diseño.	Julián Méndez	Revisión de la estructura y funcionamiento del código, de modo que cumpla con lo establecido en los diagramas de clase, actividad y componentes.	-
Evaluación de la conformidad del diseño acorde a los requerimientos .	Daniel Troyo	Revisión de cada diagrama y evaluar que cumpla los requerimientos establecidos inicialmente.	-

Realización de pruebas unitarias.	Daniel Troyo	Utilización de Junit para la evaluación de las funciones implementadas en Java, verificando por cada caso de prueba.	-
Realización de pruebas de integración.	Daniel Troyo	Utilización de Junit para la evaluación de la correcta integración de las clases de Java.	-
Realización de pruebas de aceptación.	Carlos Girón	Emulación de los casos de uso por parte del usuario del sistema completo, evaluando el módulo implementado en el sprint No se utilizará herramienta alguna..	-

Referencias

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (n.d.). *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley Professional Computing Series).