

INSTITUTO TECNOLÓGICO DE COSTA RICA

INGENIERÍA EN COMPUTACIÓN

ASEGURAMIENTO DE LA CALIDAD  
DEL SOFTWARE

---

## Avance 1

---

PROFESOR: SAÚL CALDERÓN RAMÍREZ

ESTUDIANTES:

JUAN JOSÉ GUTIÉRREZ JIMÉNEZ

KATERINE MOLINA SÁNCHEZ

ALEXANDER SÁNCHEZ BUSTAMANTE

3 DE SEPTIEMBRE DEL 2016

## Contents

<b>1 Documento de especificación de requerimientos (SyRS)</b>	<b>2</b>
1.1 Propósito del sistema . . . . .	2
1.2 Alcance del sistema . . . . .	2
1.3 Resumen del sistema . . . . .	3
1.3.1 Contexto del sistema . . . . .	3
1.3.2 Funciones del sistema . . . . .	3
1.4 Características de los usuarios . . . . .	3
1.4.1 Requerimientos funcionales . . . . .	4
1.5 Requisitos de usabilidad . . . . .	5
1.6 Requisitos de desempeño . . . . .	5
1.7 Interfaces del sistema . . . . .	5
1.8 Operaciones del sistema . . . . .	6
1.8.1 Requisitos de integración de sistemas humana . . . . .	6
1.8.2 Mantenibilidad . . . . .	6
1.8.3 Confiabilidad . . . . .	6
1.9 Estados y modos del sistema . . . . .	6
1.10 Características físicas . . . . .	7
1.10.1 Requerimientos físicos . . . . .	7
1.10.2 Requisitos de adaptabilidad . . . . .	7
1.11 Condiciones ambientales . . . . .	7
1.12 Sistema de seguridad . . . . .	7
1.13 Gestión de la información . . . . .	8
1.14 Políticas y reglamentos . . . . .	8
1.15 Sostenimiento del ciclo de vida . . . . .	9
1.16 Envasado, manipulación, envío y transporte . . . . .	9
1.17 Verificación . . . . .	9
1.18 Suposiciones y dependencias . . . . .	9
<b>2 Estándares de codificación</b>	<b>10</b>
2.1 Introducción . . . . .	10
2.2 Estándar de codificación Java . . . . .	10
2.3 Estándar de codificación Javascript . . . . .	10
<b>3 Diagramas</b>	<b>11</b>
3.1 Diagrama de componentes de sistema . . . . .	11
3.2 Diagrama de estados . . . . .	11
3.3 Diagrama de clases del sistema . . . . .	12
<b>4 Actividades de ACS</b>	<b>14</b>
<b>5 Anexos</b>	<b>15</b>
5.1 Anexo1 :Diagramas . . . . .	15
5.2 Anexo2 :Metricas de cada atributo . . . . .	17

# 1 Documento de especificación de requerimientos (SyRS)

## 1.1 Propósito del sistema

El propósito del sistema es el desarrollar una plataforma computacional automática o semi-automatizada que permita analizar diversos aspectos de desempeño del club de fútbol argentino Boca Juniors, la cual va a ser utilizada por un número de usuarios específicos(miembros de la junta directiva y director técnico).

## 1.2 Alcance del sistema

El sistema es nombrado de la siguiente manera: Analizador Automatizado de Videos del Boca Juniors, que será referido por el código AAVBJ. Este posee como alcance el satisfacer las siguientes necesidades para el Club Atlético Boca Juniors:

- Cargar un video digital almacenado en la dirección provista por el usuario.
- Automatizar la recolección de segmentos de un vídeo que cuente con información potencialmente útil.
- Poder realizar de manera automática o semiautomática la obtención de datos sobre los frames capturados de los vídeos analizados.
- Tener la capacidad de generar un informe de los resultados del análisis con la duración de cada escena detectada, la cual incluye el tiempo de inicio y tiempo final, el informe será un documento con el formato json.
- Cargar un archivo de *groundtruth* en el que el número de frame en el que sucedió un corte esté manualmente marcado, y que permita además cuantificar la cantidad de falsos positivos y negativos en la detección de cortes.
- Generar un informe de los resultados de la comparación de los dato analizados con el Groundtruth con los datos relevantes (falsos positivos y falsos negativos) analizados en el video, el informe Sera un documento con el formato json.
- El sistema no podrá realizar de manera automática o semiautomática la obtención de datos sobre la consistencia de la línea defensiva, de medio campo y ofensiva.
- Además el sistema no podrá generar un informe con las posiciones de cada jugador en un cuadro, de un equipo específico.

El sistema será capaz de poder analizar videos en alta definición de los partidos de fútbol del Club Atlético Boca Junior que juegue en el estadio sede de Alberto J. Armando (La Bombonera), los cuales proveerán información relevante que será utilizada en un análisis exhaustivo, de esta manera se obtendrán datos importantes que podrían revelar diversos aspectos de desempeño del Club Atlético Boca Juniors y sus estrategias. Con esto el Club llegará a obtener una serie de beneficios los cuales ayudarán a cumplir la meta de mejorar las estrategias del equipo, posicionamiento táctico, tanto como las habilidades individuales de los jugadores y de los entrenadores de este Club.

### **1.3 Resumen del sistema**

#### **1.3.1 Contexto del sistema**

Los elementos principales del sistema se encuentran el componente de la carga de videos, el análisis de frames, interfaz de con la información procesada y los usuarios que interactúan con el sistema, los cuales son el director técnico y los miembros de la junta directiva.

Estos elementos se interactúan de la siguiente manera: el usuario iniciará el proceso cargando el video que desea que se analice, seguidamente de que el video se suba correctamente se procede a descomponer el video por frame, así cada frame se categoriza y se le extraerá la información en necesaria para poder analizar los detalles del video al finalizar este mostrará los resultados alcanzados en la pantalla donde el usuario podrá interactuar para obtener los datos que necesita.

#### **1.3.2 Funciones del sistema**

El sistema es capaz de tomar el video que se brinde para poder examinarlo en busca de segmentos que se puedan identificar como frames válidos, los cuales se separaran para la extraer la información que se considere importante luego se analizará los datos para poder brindar una serie de resultados. Entre las limitaciones que tiene el sistema se encuentra como el sistema se desarrollara como una aplicación web, este dependerá de un conexión a internet, además el programa no sabrá con anterioridad si el video es de futbol, por lo que no restringe el tipo de video, si el video no cuenta con la calidad necesaria para el análisis este podrá presentar resultados inconsistente y erróneos.

### **1.4 Características de los usuarios**

El sistema será utilizado por un número de personas selectas que únicamente ellos poseerán los privilegios de uso de este sistema. Dentro de este grupo de personas selectas se encuentran:

- Director Técnico:Guillermo Barros Schelott.
- Presidente:Daniel Angelici.

- Vicepresidentes: Horario Paolini y Rodolfo Ferrari, Darío Richart.
- Secretarios: Gustavo Ferrari, Carlos Aguas y Pedro Orgambide.

#### 1.4.1 Requerimientos funcionales

Dentro de los requerimientos funcionales del sistema tenemos los siguientes puntos, que explicaran de manera explícita cada una de estas funcionalidades.

ID	Requerimiento	Prioridad
1.	El sistema deberá de tener la posibilidad de cargar videos digital almacenado en la dirección provista por el usuario.	Alta
2.	Visualizar cada una de las escenas detectadas (Frames) en el video cargado por el usuario.	Alta
3.	Visualizar la cantidad de cortes detectados en el video cargado por el usuario.	Alta
4.	Visualizar la energía promedio de los cortes y no cortes según g(arreglo de disimilitud entre cuadros consecutivos).	Alta
5.	Cargar un archivo de ground truth en el que el número de frame en el que sucedió un corte esté manualmente marcado, y que permita además cuantificar la cantidad de falsos positivos y negativos en la detección de cortes.	Alta
6.	El sistema deberá de comprobar la exactitud de los datos basado en un ground truth realizado anteriormente mencionado.	Alta
7.	La página web debe de contar con un sistema de muestreo de datos visible y completo.	Media
8.	Capacidad de identificar cuando sucede un error en el tiempo de ejecución.	Media
9.	Validar que los usuarios que ingresan al sistema sean usuarios con los privilegios correctos.	Media
10.	La página web del sistema deberá ser capaz de poder analizar diferentes tipo de video (extensiones).	Baja
11.	Visualizar el tiempo en procesar el vídeo cargado por el usuario.	Baja
12.	El sistema deberá de poder acoplarse a cualquier otra herramienta de análisis de datos.	Baja
13.	Capacidad de identificar cuando sucede un error en el tiempo de ejecución.	Baja
14.	Reportar los errores que se produzcan en el tiempo de ejecución del sistema.	Baja

## 1.5 Requisitos de usabilidad

Se puede indicar que el sistema sea user friendly es una de la prioridades que se toman en cuenta en el desarrollo, por lo que se definieron los siguiente requerimientos:

ID	Requerimiento	Prioridad
1.	Los usuarios debera utilizar las funcionalidades de sistema en un tipo promedio de 1.5min, sin contar el tiempo de respuesta de la pagina.	Alta
2.	Los usuarios nuevos podran aprender las funcionalidad del sistema en un tipo promedio de 3min.	Alta
3.	Realizar una página web con una tengan una estructura clara y ordenada de sus componentes.	Media
4.	Realizar una página web con un diseño que facilite la orientación del usuario durante toda la navegación.	Media
5.	Realizar una página web en la cual el contenido se debe mostrar de manera clara, con títulos comprensibles por parte del usuario.	Baja
6.	Se incluirá un manual de usuario para las consultas de cualquier funcionalidad implementada en el sistema de una manera simple que sea fácil de entender.	Baja

## 1.6 Requisitos de desempeño

Entre los requerimientos de rendimiento del sistema se abarcan los siguientes:

ID	Requerimiento	Prioridad
1.	El tiempo de respuesta de la página web del sistema debe ser notablemente bajo, este siendo menor a 0.5 segundos.	Alta
2.	La carga de una página web no deberá de consumir una gran cantidad de ancho de banda de internet.	Media
3.	Tiene la capacidad de respuesta de un mínimo de 300 usuarios.	Baja

## 1.7 Interfaces del sistema

Para la página web se utiliza un protocolo que es bastante común el Hypertext Transfer Protocol o HTTP es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. Además en el contexto de la interfaz se debe mencionar la interfaz de usuario que se va implementar la cual va a ser una página web, se describe en detalle en detalle el la sección de Requesitos de usabilidad

## 1.8 Operaciones del sistema

### 1.8.1 Requisitos de integración de sistemas humana

El sistema no cuenta con la necesidad de ser operado por algún especialista, al contrario está diseñado para poder ser utilizado por cualquier tipo de usuario e inclusive que sus conocimientos en el área sean limitadas. El único especialista que interactúan con el sistema además de los usuario convencionales son el técnico en mantenimiento de sistema.

### 1.8.2 Mantenibilidad

En el área de mantenibilidad se define los siguientes requerimientos:

ID	Requerimiento	Prioridad
1.	El sistema estará en línea un 80% del tiempo, tomando como unidad de tiempo un periodo de 365 días, lo que indica 292 días de funcionamiento..	Alta
2.	Se realizarán las actividades de mantenimiento automático de la base de datos en horas de 3:00 am a 4:00 am.	Media
3.	Acorde a la técnica del Índice de Mantenibilidad, el promedio de los índices cada módulo del sistema debe ser superior a 90.	Media
4.	Se encargará a un técnico en mantenimiento capacitado para poder operar todo lo relacionado en caso de necesitar personal para resolver alguna situación.	Baja
5.	El técnico encargado del mantenimiento se le realizará un pago de un mínimo de 12 dólares la hora.	Baja

### 1.8.3 Confiabilidad

Siendo un fallo a cualquier defecto, físico o lógico, en cualquier componente hardware o software del sistema. Existen fallos que no se tomarán en cuenta , los cuales son los siguiente: contactos accidentales entre conductores eléctricos, cortes en los mismos, defectos en los componentes, variaciones en el funcionamiento de los elementos electrónicos debidas a perturbaciones externas.

Para lo fallos no mencionados que se incluye en el sistema se tiene previsto la prevención y de ser necesario la recuperación con el objetivo de reducir la latencia de los fallos. Los cuales se categorizan como funcionamiento incorrecto de la aplicación se pueden mencionar el usuario ingresando información incorrecta, la información siendo mal procesada o la mala manipulación de los datos.

## 1.9 Estados y modos del sistema

Se adjunta en los anexos los diagramas de estados:

- Figura 2: Diagrama de generación de un GroundTruth por el usuario.

- Figura 3: Diagrama de segmentación de un video.

Este punto se desarrolla ampliamente en la Sección 3.2: Diagramas de estados

## 1.10 Características físicas

### 1.10.1 Requerimientos físicos

Al ser un sistema implementado con características web lo único necesario para darle uso a este sistema será un computador o computadores los cuales deberán ser proveídos por el club del Boca Juniors. No será necesario colocar algún tipo de servidor físico ya que el mismo será montado sobre la plataforma de Microsoft Azure, la cual es un servicio en la nube que nos ayudará a almacenar y montar la base de datos en la nube. Microsoft Azure ofrece en los niveles de servicio Básico, Estándar y Premium, se usará el servicio Estándar, el cual es el más ideal para el sistema que se está implementando, una definición de este servicio es la siguiente:

Estándar: Es la opción predilecta para la mayoría de aplicaciones en la nube, admite varias consultas simultáneas. Entre los ejemplos se incluyen aplicaciones web o para grupos de trabajo.

### 1.10.2 Requisitos de adaptabilidad

El sistema está diseñado para poder ser extendible, para poder agregar funcionalidades en un futuro, entre los requerimientos se encuentran:

ID	Requerimiento	Prioridad
1.	El sistema debe implementar patrones de diseño que permitan la adaptación de nuevas interfaces internas o externas.	Alta
2.	El sistema solo está disponible al sistema operativo de Windows.	Media

## 1.11 Condiciones ambientales

No aplica. Ninguna variable ambientales entre las cuales se menciona variables naturales, medios inducidos y medio auto-inducidos serán tomadas en cuenta en es sistema debido que el equipo ni la empresa se responsabiliza por algún daño o alteración del sistema debido a las citadas anteriormente, así se aclara que el Club Atlético de Boca Junior deberá tomar las medidas adecuadas para resolver algún problema relacionado con estas.

## 1.12 Sistema de seguridad

La seguridad de un sistema forma parte de unas de las características más relevantes de un sistema, y al ser un sistema construido para una entidad deportiva muy importante, la seguridad de este sistema debe de ser controlada de la siguiente manera:



- Proteger la información de los datos obtenida de los videos cargados por el usuario. Validar que los usuarios que ingresan al sistema sean usuarios con los privilegios correctos, en este caso solamente puede ser ingresado por los miembros de la junta directiva y el director técnico del Club Boca Juniors.
- Crear y validar que el acceso a bases de datos y robo o corrupción de datos personales o confidenciales no pueda ser realizado por ninguna persona sin los permisos necesarios.
- Crear protocolos de seguridad que permita alertar si hay alguna Intercepción de datos personales y confidenciales.
- Uso de herramientas que permitan controlar los Ataques de denegación de servicio (DoS) que deshabilitan la disponibilidad de los servicios.

### 1.13 Gestión de la información

La información del sistema va a ser manejada por medio de la interacción entre un cliente - servidor el cual va ser manejado por el Protocolo de transferencia de hipertexto ( HTTP ) el cual está diseñado para manejar este tipo de comunicaciones. Por lo tanto se van a realizar dos métodos comúnmente utilizados para una petición-respuesta entre un cliente y el servidor, estos son:

- Get - pide datos del servidor.
- Post - los datos se envían al servidor.

Además toda esta información es manejada por medio de un tipo de archivo llamado JSON (JavaScript Object Notation) el cual es un formato para el intercambios de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos que van a ser intercambiados por el cliente y el servidor. Asimismo la información que va a ser enviada y recibida por el servidor y el cliente van a seguir un protocolo de seguridad de información más especificada en la sección de seguridad del sistema.

### 1.14 Políticas y reglamentos

Se define las siguientes políticas tomadas en consideración:

- Existen solo un serie de específica de usuario que puede acceder al sistema.
- De manera automática obtener los segmentos de un vídeo con información útil.
- Generar un informe con la duración de cada escena detectada, (tiempo de inicio y tiempo final).
- De manera automática o semi-automática debe obtener datos sobre la consistencia de la línea defensiva, de medio campo y ofensiva.

- Generar un informe con las posiciones de cada jugador en un cuadro, de un equipo específico.

### **1.15 Sostenimiento del ciclo de vida**

Se destaca que durante el desarrollo del sistema se mantendrá organizado el desarrollo por medio del control de versiones, además se tendrá el estándar definido por la empresa para tener el nivel de documentación correctos sobre los el curso del desarrollo.

Además se realizarán actividades de al finalizar cada sprint con el fin de aumentar la calidad que se pueda ofrecer, estas actividades se mencionan en detalle en la sección de Actividades de ACS

### **1.16 Envasado, manipulación, envío y transporte**

No aplica. Ya que nuestro sistema no deberá de tener algún tipo de manipulación y entrega física, ya que la mayor parte de este sistema es manejado en la nube.

### **1.17 Verificación**

Las verificaciones que se realizarán al sistema ya fueron realizadas en otro documento, en dicho documento se explican cuales son las métricas que se usarán para verificar que la calidad de los elementos dentro del sistema sean el mejor, el cual se encuentra en el anexo una parte del documento, en la seccion: Metricas

### **1.18 Suposiciones y dependencias**

Dentro de las suposiciones contamos con los siguientes puntos:

- El club del Boca Juniors contará con el equipo necesario para el uso del sistema dentro de su organización.
- El club del Boca Juniors cuenta con el servicio ancho de banda de internet.
- El formato del video que es cargado por el usuario cumple con los requisitos necesarios para ser leído.
- Se contará con una base de datos sobre la plataforma de Microsoft Azure para contar con un registro de los datos analizados por los vídeos cargados por los usuarios.
- El club del Boca Juniors cuenta con los recursos necesarios para hacer los pagos de los servicios de Microsoft Azure.

Dentro de las dependencias contamos con los siguientes puntos:

- El uso del sistema dependerá de un servicio de ancho de banda de internet estable y rápido.

- El uso del sistema dependerá de la estabilidad del servicio de Microsoft Azure.
- El club del Boca Juniors debe de contar con el servicio eléctrico estable necesario para el funcionamiento de las computadoras que utilizan el sistema.

## 2 Estándares de codificación

### 2.1 Introducción

Debido a la necesidad de mantenimiento del código presente en la mayoría de proyectos de software, surge la importancia de seguir un estándar de codificación para garantizar una modificación fácil y accesible para el desarrollador que analizará en el futuro el código, el cual en la mayoría de ocasiones será una persona distinta del escritor inicial. Por lo tanto, el equipo de sport analytics deberá seguir los siguientes estándares de codificación presentados a continuación.

### 2.2 Estándar de codificación Java

El estándar escogido para la elaboración del proyecto para Boca Juniors corresponde al establecido por la empresa Google, el cual declara ampliamente la forma de codificación en el lenguaje Java. La guía propuesta toma aspectos no encontrados en otros estándares como lo son los try and catch o Javadoc, los cuales son importantes para lograr medir por ejemplo la documentación interna del código. El estándar de Google se adapta a las necesidades del proyecto debido a que sigue principios de código limpio como lo es la declaración adecuada de variables con nombre significativos, además de contener prácticas comunes entre los programadores java lo que facilita la adaptación al estándar. Por lo tanto, al seguir esta guía de codificación se garantiza la calidad del código del producto para su posterior mantenimiento.

Referencia: Google Java Style Guide

### 2.3 Estándar de codificación Javascript

En la búsqueda del estándar a seguir para la codificación en javascript se analizaron las guías más populares incluyendo la presentada por Google y Crockfords. En comparación a todas las observadas Airbnb se distingue debido a su completitud y facilidad de aprendizaje del estándar. Los ejemplos encontrados en el repositorio en el cual se encuentra el estilo, muestran explícitamente la forma incorrecta de escribir el código adjuntando seguidamente la correcta, lo que simplifica identificar si el desarrollador está realizando la codificación correctamente, siendo esta característica la más relevante para la decisión tomada. Además, este estándar utiliza como referencia en su elaboración el estilo de Google e Idiomatic JavaScript, en este último se puede obtener referencia para la búsqueda de frameworks que permitan la generación de test unitarios como

lo son QUnit o JsTestDriver.  
Referencia: Airbnb JavaScript Style Guide

## 3 Diagramas

### 3.1 Diagrama de componentes de sistema

En este diagrama de componentes se representa la distribución y conexión entre los componentes que integran el sistema para Boca Juniors. Del parte del cliente, debido a la naturaleza de la aplicación web, únicamente se contiene el navegador por el cual se accede a la aplicación. Por otro lado, dentro del servidor se contendrá el web service que el cliente podrá consumir para lograr segmentar un video. Este service ofrecerá la página de interfaz index.jsp donde se realizarán a través de angularJS las peticiones Get y Post correspondientes para mostrar una aplicación dinámica. Además, se accederán a los servlets por el mismo service para el procesamiento del video mediante openCV. Se crea un componente especial para el ingreso de los usuarios autorizados. Como último componente relevante del diagrama se coloca la base de datos, la cual se encontrará dentro del mismo servidor.

Se anexa la imagen con el diagrama:

Diagrama de componentes de sistema

### 3.2 Diagrama de estados

El objetivo principal de realizar los diagramas de estados es apreciar en las etapas principales en los que se encontrará el sistema para obtener información relevante para la elaboración del diagrama de clases y sus relaciones. El sistema tendrá dos estados principales, el primero será la obtención del video y su procesamiento para la apreciación visual de los cortes generados por el algoritmo de segmentación. El segundo estado principal, será la creación de un archivo groundTruth con extensión YAML para su posterior comparación con los datos generados por la segmentación automática del sistema. Por ende, a continuación se mostrarán dos diagramas que mostrarán los estados antes mencionados.

Se anexa la imagen con el diagrama:

Diagrama de generación de un GroundTruth por el usuario.

El usuario al seleccionar un video coloca al sistema en un estado de reproducción del mismo, donde está acción es pausada para marcar un corte en el video reproducido. Después el sistema genera el archivo. Seguidamente, se mostrará el archivo generado y este escogerá si sólo se guardará o comparará con el archivo generado por el segmentador.

Se anexa la imagen con el diagrama:

### Diagrama de segmentación de un video

En este diagrama se logra apreciar un flujo condicional donde si el archivo no es correcto el sistema termina la acción. Seguidamente se aprecian estados secuenciales, y en la última parte del diagrama se divide un flujo paralelo donde el sistema envía los datos al usuario para ser mostrados, guarda el archivo generado en memoria dinámica en la base de datos, y genera el informe del archivo segmentado.

### 3.3 Diagrama de clases del sistema

El diagrama de clases representa en su mayoría la parte del Back-end que se implementará para el proyecto. Se utiliza el patrón adaptador que establece los requisitos que tiene que cumplir una librería de edición de video para lograr adecuarse y segmentar algún video dado. También se crea el comportamiento del algoritmo de segmentación que es implementado por el videoEditor debido a que este es el que provee el comportamiento del segmentador, esta interfaz también permite obtener un mayor grado de cohesión entre los métodos existentes en el segmentador. En la clase segmentadorTemporal, tenemos los atributos que se necesitaran para implementar correctamente la edición del video. Se mantiene el estilo MVC, por lo cual se crean controladores como capa intermedia entre la conexión con el usuario. Se refleja explícitamente la conexión con los servlets que proveerán a la interfaz los datos los servicios suficientes para mostrar al usuario. Sólo se representa el nombre de las librerías que se utilizarán al igual que la conexión con angular JS.

Se anexa la imagen con el diagrama:

Diagrama de clases del sistema



## 4 Actividades de ACS

Actividad	Responsable	Sprint
Revisión de la construcción de los requerimientos del sistema establecidos con el estándar ISO/IEC/IEEE 29148-2011.	Alexander Sánchez B, Juan Jose Gutierrez J, Katherine Molina S	Sprint 1.
Revisión y análisis de los requerimientos del cliente.	Juan José Gutiérrez J	Sprint 1 y 2.
Revisión de la construcción de los requerimientos del sistema basado en los atributos del ISO-9126.	Alexander Sánchez	Sprint 1.
Evaluación de la implementación de código que siga los estándares de codificación.	Katherine Molina S, Alexander Sanchez B.	Sprint 1 y 2
Revisión y evaluación de la herramienta de control de cumplimiento de estándares de codificación.	Alexander Sánchez B, Juan Jose Gutierrez J, Katherine Molina S	Sprint 2.
Identificar los objetivos que cumplen el estándar de código según la orientación del sistema.	Alexander Sánchez B, Juan Jose Gutierrez J	Sprint 2.
Valoración de los principios de diseño acorde al diagrama de clases.	Alexander Sanchez B	Sprint 1 y 3.
Evaluación de la relación del diseño de componentes según los requerimientos planteados.	Juan José Gutiérrez J	Sprint 1 y 3.
Construir una validación de ingreso de usuarios válidos del sistema.	Juan José Gutiérrez J	Sprint 2.
Analizar los riesgos técnicos que se pueden experimentar.	Katherine Molina S	Sprint 3.
Inspeccionar la integración de los patrones de diseño implementados.	Alexander Sánchez B, Juan Jose Gutierrez J	Sprint 3.
Revisión y evaluación de los diagramas UML que detallan la arquitectura del sistema.	Katherine Molina S	Sprint 2.
Crear pruebas para estresar el componente a su máximo nivel.	Alexander Sanchez B	Sprint 3.
Construir un conjunto de pruebas unitarias para la validación de funcionamientos específicos del sistema.	Katherine Molina S, Alexander Sanchez B	Sprint 1, 2 y 3.
Revisar y analizar las respuestas obtenidas de las pruebas unitarias del sistema.	Alexander Sánchez B, Juan Jose Gutierrez J	Sprint 1, 2 y 3.
Construir un conjunto de pruebas de integración para la validación de módulos en conjunto del sistema.	Alexander Sánchez B, Juan Jose Gutierrez J Katherine Molina S	Sprint 3.
Revisar y analizar las respuestas obtenidas de las pruebas integración del sistema.	Katherine Molina S, Alexander Sanchez B	Sprint 3.
Construir un conjunto de pruebas de aceptación para la validación de la funcionalidad completa del sistema.	Alexander Sánchez B, Juan Jose Gutierrez J, Katherine Molina S	Sprint 3.
Revisar y analizar las respuestas obtenidas de las pruebas de aceptación del sistema.	Alexander Sanchez B	Sprint 4.
Realizar una revisión de los tiempos de respuesta de la página web sean bajos.	Katherine Molina S, Juan Jose Gutierrez J, Alexander Sanchez B	Sprint 4.

## 5 Anexos

### 5.1 Anexo1 :Diagramas

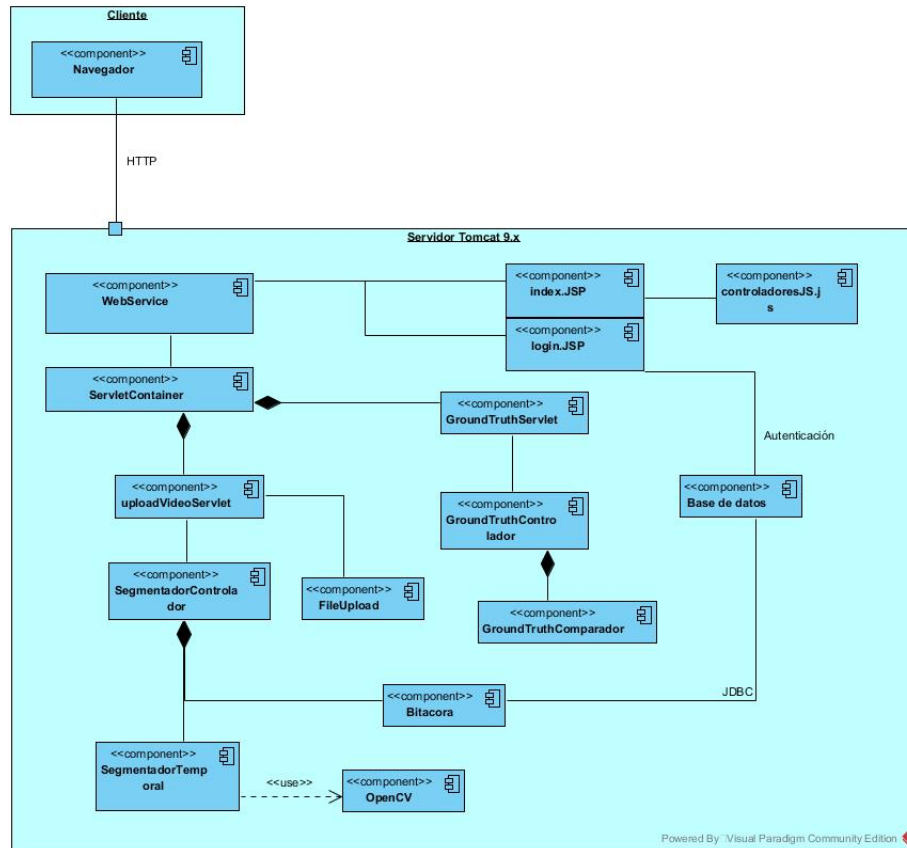


Figure 1: Diagrama de componentes de sistema



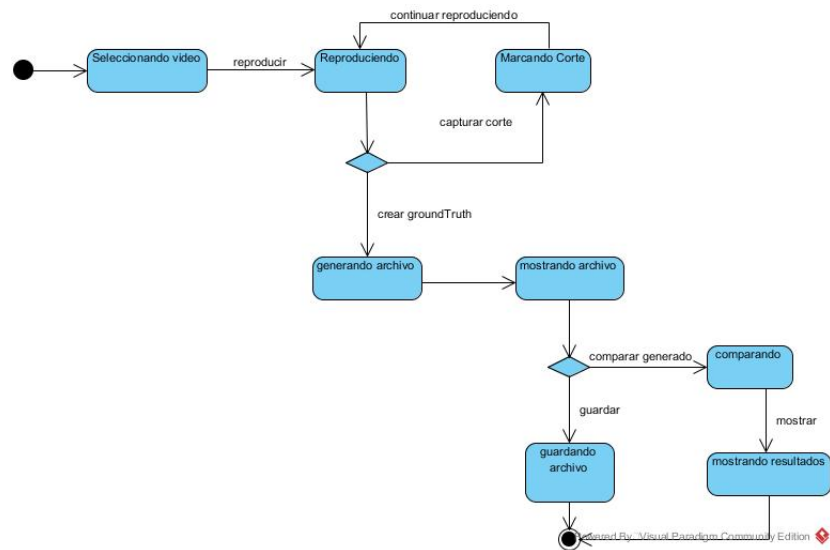


Figure 2: Diagrama de generación de un GroundTruth por el usuario.

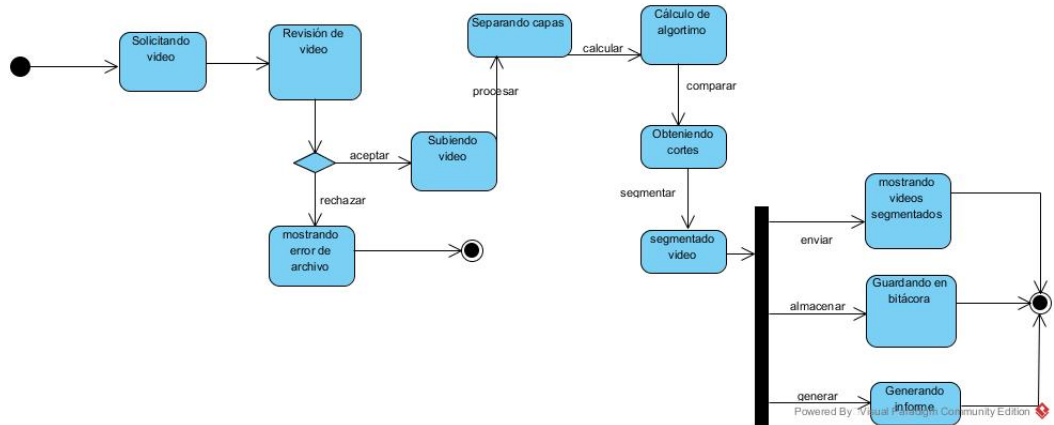


Figure 3: Diagrama de segmentación de un video

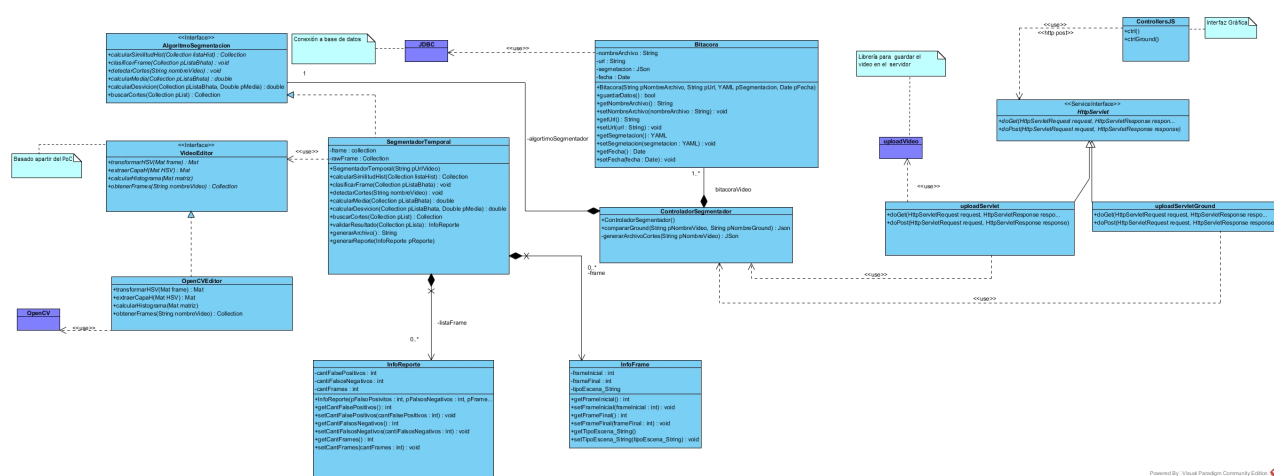


Figure 4: Diagrama de clases del sistema

## 5.2 Anexo2 :Metricas de cada atributo

Funcionalidad			
Atributos	Metrica	Nivel requerido	Herramienta
Seguridad	Interna: Analiza el código fuente del sistema para detectar vulnerabilidades.	No deben existir ningun vulnerabilidad crítica.	Fortify software
Adecuación	Interna: Evaluar el promedio de la cantidad de tareas asignadas según 3 requerimientos más relevantes del sistema.	Deben existir 3 o más tareas relacionadas con el requerimientos.	Zoho, ReqView
Exactitud	Interna: Se obtiene el porcentaje según la cantidad de frames correctos en un conjunto de 50 frames.	El porcentaje debe ser mayor al 85%.	GroundTruth
Eficiencia			
Comportamiento en el tiempo	Externa: El tiempo promedio de respuesta que tarda al solicitudes http a la página web del sistema cuando solicitar las posiciones de los jugadores.	Es tiempo promedio debe ser menor a un segundo.	Jmeter
Utilización de recursos	Interna: La cantidad de líneas de código utilizadas al generar el informe de las escenas detectadas.	La cantidad de líneas debe ser inferior a 300.	EclEmma
Conformidad del rendimiento	Analizar el código fuente para encontrar olores de software y pulgas.	No deben existir ningun error clasificado como crítico o mayor.	Sonarqube
Usabilidad			
Facilidad de Aprendizaje	Externa: Una de las métricas de calidad externas en la facilidad de aprendizaje es hacer una medición de los tiempos que duran los miembros de la junta directiva del Boca Juniors por cada una de las funcionalidades de la página web por primera vez.	Los tiempos de aprendizaje sean bajos, entre un rango de 10 a 20 minutos.	Cualquier herramienta con un cronómetro.
Operabilidad	Externa: Una de las métricas de calidad externas en la operabilidad es hacer una encuesta a los miembros de la junta directiva del Boca Juniors por cada una de las funcionalidades de la página web por primera vez.	Que haya un nivel de satisfacción alto(Rango del 1 al 10) en la operabilidad.del sistema	Cualquier herramienta para hacer encuestas

Portabilidad			
Atributos	Metrica	Nivel requerido	Herramienta
Adaptabilidad	Externa: Una de las medidas de calidad externas en la adaptabilidad es que todos los miembros de la junta directiva tengan la facilidad de poder visualizar las funcionalidades del sistema desde su computadoras, en este caso la aplicación web. Por lo tanto que puedan usar cualquier navegador web y puedan visualizar estas funcionalidades con facilidad.	Visualización de la aplicación en todos los navegadores web.	Cualquier navegador web: Google Chrome, Firefox, Safari, entre otros.
Capacidad de instalación	Externa: Una de las medidas de calidad externas en la capacidad de instalación es ver la facilidad en que los miembros de la junta directiva pudieron instalar el sistema en sus computadoras personales.	Poder instalar el sistema sin ningún tipo de problema, en un tiempo de no mas de 10 minutos.	Cualquier herramienta de encuestas
Coexistencia	Interna: Una de las medidas de calidad internas en la coexistencia es que el servidor de la página web pueda funcionar con una herramienta que pueda detectar y bloquear cualquier tipo de ataque DDOS.	Que la coexistencia de otras herramientas afecten el funcionamiento del sistema.	Log2ban es una herramienta que permite la detección y bloqueo de IPs que participan en ataques DDOS.
Fiabilidad			
Tolerancia a fallos	Externa: Maneja correctamente los fallos, mostrando al usuario el error o procesa los internamente manteniendo el sistema con otros funcionamientos habilitados.	El sistema no cierra su ejecución, únicamente si el usuario así lo desea.	Junit
Madurez	Externa: El sistema presenta mínimas fallas durante su ejecución	No se generan problemas en la utilización del sistema	Junit
Cumplimiento de fiabilidad	Externa: El sistema no alberga ningún dato relacionado con técnicas seleccionadas (Contexto Deportivo) sobre los vídeos analizados. Los videos son únicamente analizados, no almacenados ni distribuidos (Copyright).	El ámbito de derechos de los videos pueden variar, nivel requerido es 50%	Investigación de documentación

<b>Mantenibilidad</b>			
Estabilidad	Externo: El sistema mantiene su integridad si se modifica algún componente. Interno: La implementación presenta alta cohesión y bajo acoplamiento entre sus componentes	La modificación no afecta el funcionamiento de otros componentes.	Sonarqub
Cambiabilidad	Interno: El diseño implementado cumple con los principios SOLID relacionados con la programación orientada a objetos.	Se observa la posibilidad de extensión del diseño para la integración de nuevos requerimientos. Justifica la violación de cierto principio establecido.	Visual Paradigm 13.1
Análisis	Interno: El código escrito presenta los siguientes olores relacionados con software: Código duplicado o “muerto”, poca cohesión a nivel de clases y paquetes, alto acoplamiento, escritura ambigua, poca documentación.	El código presente cumple con la mayoría de buenas prácticas de programación y justifica el incumplimiento de las que no cumple.	Sonarqub