

Avance 1: Proyecto Semestral

Valeria Calderón, Lucy Chaves y Seth Stalley

3 de septiembre de 2016

En este avance, se presenta un documento de requerimientos basado en los atributos de calidad escogidos para el estándar ISO-9126, además se fija el estándar de codificación a utilizar a lo largo del proyecto.

Índice

1. Documento de Requerimientos	3
1.1. Propósito	3
1.2. Alcance	3
1.3. Descripción General	3
1.3.1. Contexto	3
1.3.2. Funciones	4
1.3.3. Características del usuario	4
1.4. Requerimientos Funcionales	5
1.5. Requerimientos de Usabilidad	5
1.6. Requerimientos de Desempeño	6
1.7. Interfaces del Sistema	7
1.8. Operaciones del sistema	9
1.8.1. Requerimientos de integración del sistema humano	9
1.8.2. Mantenibilidad	9
1.8.3. Fiabilidad	10
1.9. Modos y estados del sistema	10
1.10. Características físicas	12
1.10.1. Requerimientos físicos	12
1.10.2. Requerimientos de adaptabilidad	12
1.11. Condiciones de Ambiente	12
1.12. Seguridad del sistema	12
1.13. Administración de la información	13
1.14. Políticas y regulaciones	13
1.15. Sostenimiento del ciclo de vida del sistema	13
1.16. Empaquetado, manejo, envío y transporte	14
1.17. Verificación	14
1.18. Suposiciones y dependencias	14

2. Estándar de codificación	15
2.1. Java	15
2.1.1. Propiedades destacadas del estándar:	15
2.2. Angular 2.0 (Typescript)	16
2.2.1. Propiedades destacadas del estándar	16
3. Diagramas	16
3.1. Componentes	17
3.2. UML	18
3.2.1. Front-End	18
3.2.2. Back-End	18
4. Actividades de Aseguramiento de la Calidad del Software	19
4.1. Planeamiento del proceso de ACS	19
4.2. Aseguramiento del producto	19
4.3. Aseguramiento del proceso	20
5. Referencias	21

1. Documento de Requerimientos

1.1. Propósito

El sistema, que trabaja a partir de un vídeo digital, realiza una segmentación y clasificación por equipo de los jugadores en un partido de fútbol. El objetivo general del programa es que los miembros del club del Boca Juniors, constituido por el director técnico y la junta directiva, puedan realizar análisis de forma automática o semiautomática sobre el desempeño y rendimiento que poseen los jugadores y el club en general. Además, el sistema genera un informe sobre la cantidad de jugadores que aparecen en cada cuadro, de esta forma el club Boca Junior va a poder obtener datos y realizar análisis sobre la consistencia de la línea defensiva, medio campo y ofensiva.

1.2. Alcance

1. **Nombre:** Sistema de segmentación y clasificación por equipo de los jugadores de un partido de fútbol en un vídeo digital
2. **Necesidades del usuario y lo que el sistema realiza para satisfacerlas:** La junta directiva, así como el cuerpo técnico desean realizar estudios sobre el desempeño del club y los jugadores, por esta razón desean obtener datos con base a un vídeo que les permita hacer los análisis respectivos sobre la consistencia de la línea defensiva, medio campo y ofensiva. El sistema será capaz de procesar los vídeos que desean analizar, además tiene la capacidad de mostrar y descargar información como la cantidad de jugadores por equipo en cada cuadro, el vídeo mostrando los blobs, que representan los jugadores, y sus respectivas etiquetas que representan un equipo. El programa también va a realizar una métrica de exactitud de los resultados para poder verificar la veracidad de los datos obtenidos.
3. **Beneficios, objetivos y metas:** El beneficio principal del sistema es la obtención automática o semiautomática de información relevante encontrada en un vídeo de un partido de fútbol. Poder utilizar la información obtenida para el análisis sobre la consistencia de las líneas del equipo del Boca Juniors es el principal objetivo del sistema, así como de la junta directiva y el cuerpo técnico. Por último la meta del sistema es brindar la información relevante como lo es el vídeo con los blobs marcados con la etiqueta correspondiente a su equipo, la cantidad de jugadores por equipo encontrados en cada cuadro y la generación de informes basados en los elementos obtenidos del vídeo.

1.3. Descripción General

1.3.1. Contexto

La realización de un sistema que permita analizar diversos aspectos sobre el desempeño del club el Boca Juniors surgió de la necesidad del presidente del

club, Daniel Angelici junto con los demás miembros de la junta directiva, de realizar estudios sobre el desempeño mostrado en el campo por parte de los jugadores al momento de un partido.

Una de las principales funciones del sistema es la generación de informes, esta característica permita que los miembros del club y el director técnico del equipo puedan mantener una bitácora con resultados de varios partidos y de esta manera analizar aquellos errores que son reincidentes, cuales son nuevos y en que aspectos mejorar la técnica del equipo.

Las únicas personas que pueden ingresar al sistema, para solicitar el procesamiento de un vídeo, la generación de informes o cualquier otra funcionalidad, son los miembros de la junta directiva y el director técnico de la selección mayor del Boca Juniors

1.3.2. Funciones

El sistema posee varias funcionalidades de las cuales destacan la posibilidad de cargar un vídeo digital que se encuentre almacenado en una dirección dada por el usuario, realizar sobre el vídeo una clasificación por equipo de los jugadores y convertir las siluetas de los jugadores en blobs. También da la posibilidad de descargar el vídeo con los blobs (los blobs son manchas en una imagen que representan los jugadores) y las etiquetas por equipo marcados. El programa además va a generar un informe de la cantidad de jugadores de cada equipo en cada cuadro, el informe se devolverá en un archivo con formato csv.

Mientras el vídeo corre, el programa va a permitir la visualización del número de jugadores detectados en cada cuadro, además de poder ver los blobs y las etiquetas por equipo en cada cuadro. El tiempo que duró el procesamiento del vídeo también se podrá conocer.

Por último, el sistema tiene la posibilidad de cargar un archivo de "Ground Truth", donde están los datos correctos sobre los blobs en cada cuadro, esto permite cuantificar la cantidad de fallos que se den en la clasificación no supervisada.

1.3.3. Características del usuario

- Junta directiva y director técnico
 - Función: Usuario
 - Número: 8
 - Uso del sistema: Subir vídeos al sistema y poder obtener datos que permitan realizar un análisis del desempeño de los jugadores del Boca Juniors.
- Encargado de mantenimiento
 - Función: realizar mantenimiento del sistema
 - Numero: 1

- Uso del sistema: Solucionar pulgas del sistema, instalar actualizaciones, resolver problemas que hayan surgido debido a errores de usuario.

1.4. Requerimientos Funcionales

La siguiente lista son los requerimientos funcionales del sistema,

- REQ 1 – El usuario deberá tener la posibilidad de cargar un vídeo al sistema, para lograrlo el usuario debe proveer la dirección donde el archivo se encuentra localizado.
- REQ 2 – El usuario deberá poder realizar descargas del vídeo que contenga los blobs, además los blobs deben tener una etiqueta que permita diferenciar que pertenecen a diferentes equipos.
- REQ 3 – El sistema deberá generar un archivo en formato csv que contenga un informe con la cantidad de jugadores, clasificados por equipo, que aparecen en cada recuadro.
- REQ 4 – El sistema permitirá la visualización de la cantidad de jugadores que se detecten en cada cuadro durante toda la duración del vídeo.
- REQ 5 – El sistema debe permitir la visualización del vídeo donde se muestren los “blobs”, que se define como las regiones que representan un jugador
- REQ 6 – El sistema deberá desplegar las etiquetas que se definieron por cada equipo junto a los blobs que corresponden a los jugadores durante el transcurso de todo el vídeo.
- REQ 7 – El sistema deberá mostrar el tiempo que tardó el procesamiento del vídeo completo.
- REQ 8 – El sistema debe permitir cargar un archivo de “Ground Truth”.
- REQ 9 – El sistema debe ser capaz de realizar el conteo de la cantidad de fallos que la clasificación no supervisada tuvo con respecto al archivo de “Ground Truth” que se cargó.

1.5. Requerimientos de Usabilidad

Ahora se despliegan los requerimientos de usabilidad, que son aquellos encargados de medir la efectividad, eficiencia y satisfacción del sistema en el contexto que es necesitado

- REQ 10 – El sistema deberá ser fácil de comprender para el usuario, es decir el usuario debe entender el comportamiento y manejo del sistema.

- Métrica: Entendimiento de las funciones, el promedio los usuarios deben entender la forma de ejecución y objetivo de al menos el 90 % de las funciones totales del sistema.
- REQ 11 – El sistema deberá ser fácil de manejar, es decir que el usuario con sencillez pueda realizar una tarea.
 - Métrica: Cantidad de clicks para realizar una tarea, es el promedio de clicks que les tome a los usuarios aprender a usar una función específica del sistema debe ser en promedio menor a 5 clicks
- REQ 12 – El sistema debe ser fácil de operar por el usuario
 - Métrica: Tiempo de errores entre usuario, se saca obteniendo el tiempo que hay entre cada error humano que se manifieste debe ser en promedio mayor a los 3600 segundos (1 hora)
- REQ 13 – El sistema debe ser capaz de explicar porque se da una falla y que el usuario sea capaz de entenderlo. Por ejemplo el archivo que cargo no fue compatible, el usuario no tiene los permisos para acceder al sistema.
 - Métrica: Entendimiento de las fallas, en caso de surgir un error, al menos en el 90 % de las ocasiones el usuario debe ser capaz de explicar porque motivo sucedió el error.

1.6. Requerimientos de Desempeño

A continuación se listan los requerimientos de desempeño del sistema.

- REQ 14 – El sistema debe procesar los vídeos de 10 segundos, en menos de 5 segundos.
 - Métrica: Duración del procesamiento, el procesamiento de un vídeo, que tenga una duración aproximada a los 10 segundos, debe ser menor a los 5 segundos en al menos 95 % de los casos.
- REQ 15 – El sistema debe generar los informes csv en menos de 2 segundos
 - Métrica: duración en generación de informes, el tiempo de generación del informe en formato csv debe ser menor a los 2 segundos en el 95 % de los casos.
- REQ 16 – El sistema deberá obtener las métricas de exactitud en menos de 5 segundos.
 - Métrica: tiempo de obtención de métrica de exactitud, el tiempo de obtención de resultados con respecto al análisis y la cuantificación que se realiza entre el archivo de "Ground Truth" y la función de la clasificación no supervisada, debe ser menor a 5 segundos en al menos el 95 % de los casos.

- REQ 17 – El sistema debe cargar un vídeo en menos de 2 segundos.
 - Métrica: Tiempo de carga de un vídeo, la cantidad de tiempo que el sistema debe durar para cargar un vídeo después de que el usuario haya especificado la dirección para localizarlo, debe ser menor a los 2 segundos en el 95 % de los casos.

1.7. Interfaces del Sistema

Interfaces de Usuario

El sistema de segmentación y clasificación de jugadores va a tener las siguientes pantallas como las encargadas de la interacción entre el sistema y el usuario.

La primera pantalla será la de «log in» o acceso al sistema, que tendrá dos campos para rellenar, uno con el nombre de usuario y otro con la contraseña del usuario, solo las personas que formen parte de la junta directiva y el director técnico recibirán nombres de usuario y contraseña para entrar al sistema.

La segunda pantalla posee un espacio para subir los vídeos que se quieran analizar, ya sea arrastrándolos a la pantalla o cargándolos, el vídeo se analizará una vez que haya sido subido

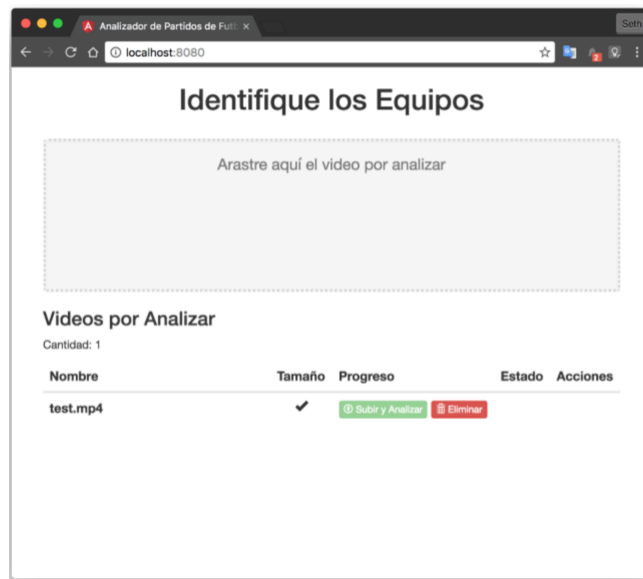


Fig 1. Pantalla de inicio del sistema

La tercera pantalla contendrá el vídeo con los resultados de los blobs, además mostrará información importante como la cantidad de jugadores por equipo en cada cuadro.



Fig 2. Muestra de los posibles resultados del sistema

Por último se tendrá una pantalla que permita que se suba al sistema un archivo de "Ground Truth" que además va a ser analizado para verificar la veracidad de los resultados del sistema. Estos resultados se analizarán y se mostrarán en esta pantalla.

Interfaces de Software

Para el sistema de segmentación y clasificación de jugadores por equipo, que se realiza para el club argentino el Boca Juniors, se desarrollará el "Backend" del sistema utilizando Java como lenguaje de programación, es importante mencionar que se agregó la librería de OpenCV para facilitar el manejo de los videos a procesar.

Para el "Front-End" se realiza una interfaz web usando AngularJs 2.0, el acceso al sistema va a ser por medio de una página web que se encargará de la comunicación entre el usuario y el "Back-End" del sistema.

Interfaces de Hardware

El usuario que desee tener acceso al sistema de segmentación y clasificación de jugadores por equipo, debe contar con una computadora y tener conexión a internet, de esta forma podrá ingresar a la página web del sistema y poder utilizar sus funciones.

1.8. Operaciones del sistema

1.8.1. Requerimientos de integración del sistema humano

El sistema está pensado para que se uso se fácil para cualquier usuario con poca o mucha experiencia en este tipo de programas, así que no es necesario algún tipo de personal capacitado para el uso del sistema. Sin embargo los usuarios van a recibir un manual donde estén especificadas y la forma de ejecución de las tareas en el sistema.

La única área que necesita una persona con experiencia para manejar el sistema es el área de mantenimiento, pues cualquier error que cometa puede sabotear el sistema.

1.8.2. Mantenibilidad

La siguiente lista representa los requerimientos de mantenibilidad del sistema.

- REQ 18 – El sistema debe tener código ordenado, sin presencia de olores de software.
 - Métrica: Complejidad ciclomática, la complejidad ciclomática del código debe ser menor a 7.
- REQ 19 – Las funciones del sistema deben realizar una única tarea, ser sencillas.
 - Métrica: Complejidad de los métodos, los métodos que forman parte del código del sistema no pueden exceder las 15 líneas de código.
- REQ 20 – El sistema debe ser de fácil entendimiento para la persona que le de mantenimiento de forma que pueda darse cuenta de donde provienen los errores.
 - Métrica: Análisis de errores: El promedio de la cantidad de errores que no es posible determinar de dónde provienen o porque se están manifestando debe ser menor al 6 %.
- REQ 21 – El sistema debe poseer una bitácora de cambios.
 - Métrica: Registro de cambios, el promedio de la cantidad de cambios que se le han realizado al sistema y han sido debidamente registrados debe ser al menos de 90 %.
- REQ 22 – El sistema debe ser capaz de manejar que se le realicen modificaciones sin ningún problema.
 - Métrica: Impacto de las modificaciones, el promedio de veces que sistema falle después de haberle realizado algún tipo de modificación debe ser menor al 5 %.

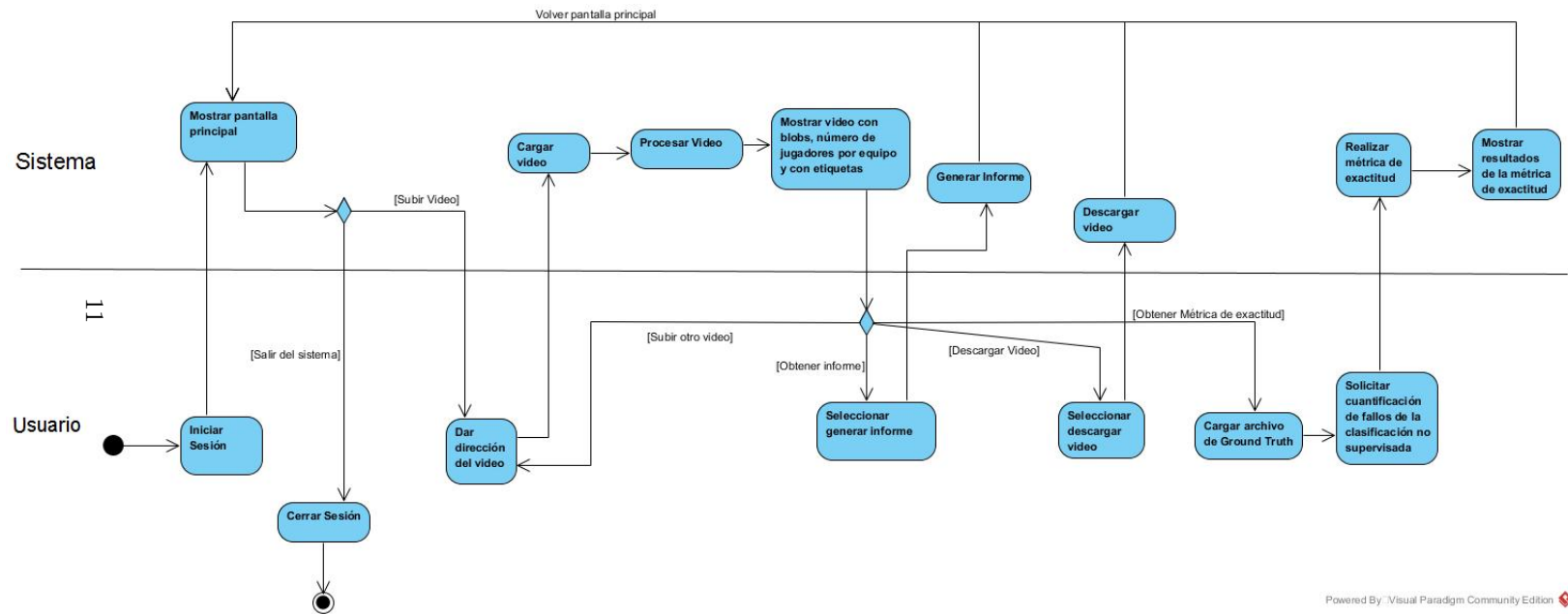
1.8.3. Fiabilidad

Los requerimientos de Fiabilidad son los siguientes.

- REQ 23 – El sistema no debe tener fallas.
 - Métrica: Detección de fallas, el promedio de los elementos que causan fallas debe ser menor al 5 %.
- REQ 24 – El sistema debe ser capaz de no caerse en caso de recibir un archivo no permitido.
 - Métrica: Manejo de parámetros incorrectos, el promedio de funciones que tengan la capacidad de manejar parámetros no permitidos, es decir cuando el usuario carga un archivo no permitido y el sistema es capaz de seguir trabajando e informarle al usuario sobre el error, debe ser mayor al 90 %.
- REQ 25 – El sistema no puede estar inhabilitado por mas de 30 segundos
 - Métrica: Tiempo inhabilitado, el promedio de tiempo que el sistema pueda estar fuera de servicio después de una falla debe ser menor a 30 segundos.

1.9. Modos y estados del sistema

El siguiente diagrama de estados muestra todos los modos del sistema:



1.10. Características físicas

1.10.1. Requerimientos físicos

El sistema va a ser instalado en las computadoras de las oficinas de los miembros de la junta directiva y el director técnico del Boca Junior, el programa debe trabajar correctamente en los diferentes ambientes físicos a los que va a ser expuesto, por esta razón no tiene ningún tipo de restricción con respecto a peso, volumen o dimensión sobre el área donde va a ser instalado, tampoco es necesario la compra de materiales o la contratación de personal para la construcción de un entorno aceptable para el sistema.

1.10.2. Requerimientos de adaptabilidad

Los requerimientos de adaptabilidad son los que definen el crecimiento, capacidad, contracción y expansión del sistema, en este caso se presentan los siguientes.

- REQ 26 - El sistema deberá ser adaptable a otros ambientes
 - Métrica: Adaptabilidad de ambientes del sistema, es el promedio de cambios a realizar en funciones al sistema cada vez que se quiera instalar el sistema en un nuevo ambiente

1.11. Condiciones de Ambiente

El sistema va a trabajar con vídeos que se hayan grabado en el estadio del Boca Juniors, conocido como La Bombonera, en las imágenes que se graben es posible encontrar elementos como viento, lluvia, luz solar y artificial, zacate las cuales van a ser debidamente manejadas por el sistema. Elementos como el ruido y la temperatura no afectaran el rendimiento ni los resultados que el programa proporcione. En ámbitos de organización propios del club, el sistema solo se puede ser utilizado por miembros de la junta directiva del club o el director técnico. Y ninguna otra persona, empresa, club puede tener acceso a los resultados que el sistema aporte de acuerdo a los análisis que realizó.

1.12. Seguridad del sistema

Requerimientos de seguridad:

- REQ 27 – El sistema deberá permitir el acceso a las funcionalidades únicamente a aquellas personas que sean miembros de la junta directiva del club el Boca Juniors y al director técnico de dicho club.
 - Métrica: Control de acceso, el promedio de personas indebidas que ingresaron al sistema debe ser menor al 3 %.

1.13. Administración de la información

Los siguientes requerimientos son sobre el manejo de la información que realiza el sistema

- REQ 28 – El sistema debe ser capaz de poder cargar vídeos en formato .mp4, es el único tipo de archivo y formato que el sistema es capaz de procesar.
- REQ 29 – El sistema debe ser capaz de generar un archivo en formato csv que contenga la cantidad de jugadores por equipo que aparezca en cada cuadro. Este archivo no se almacena en el sistema, sin embargo se descargará a la computadora que se este utilizando.
- REQ 30 – El sistema debe ser capaz de exportar un vídeo en formato mpeg que contenga los blobs y las etiquetas de los jugadores. El vídeo no se almacenará en el sistema.

1.14. Políticas y regulaciones

Ya que el sistema no recibe ningún tipo de información personal de los usuarios no existen ninguna política o regulación con respecto a la protección de la información personal. Además no se impusieron políticas provenientes de agencias regulatorias.

En el área de salud y seguridad no existe problema alguno del sistema con respecto al uso de elementos que puedan ser tóxicos o generar radiaciones electromagnéticas que puedan dañar a las personas que lo utilicen.

Es importante el cumplimiento de la exclusividad con el club Boca Juniors, es decir solo este club tiene acceso al sistema creado por la empresa Sport Analytics.

1.15. Sostenimiento del ciclo de vida del sistema

El ciclo de vida logístico del sistema se da después de las negociaciones del club Boca Juniors con la empresa Sport Analytics, una vez que se entiende lo que el cliente desea se realizan los siguientes pasos:

- Proveer estrategias y sistemas que puedan satisfacer las necesidades de los requerimientos del sistema.
- Realizar el proceso de diseño para obtener un sistema que sea sostenible y adecuado para las necesidades del usuario.
- Manejar congruencia entre los requerimientos, el diseño realizado y la implementación del sistema
- Dar capacitaciones, a los desarrolladores, sobre las herramientas a utilizar durante el procesos administración del proyecto, la implementación del mismo y el testeo.

1.16. Empaquetado, manejo, envío y transporte

El sistema va a ser desarrollado en Costa Rica en las provincias de San José y Cartago, sin embargo su utilización se va a dar en Buenos Aires, Argentina.

Para trasladar el software de Costa Rica a Buenos Aires la junta directiva del Boca Juniors solicito que la versión final fuera entregada en una llave maya, Sport Analytics proveerá un ingeniero encargado de llevar el sistema para Argentina, el ingeniero será recibido por un dirigente del Boca Juniors. Ambos se encargarán del transporte hacia las oficinas del club. EL ingeniero deberá realizar las instalaciones del sistema y quedarse en Argentina hasta que el programa trabaje con normalidad.

1.17. Verificación

La siguiente lista muestra los métodos planeados para la verificación del sistema:

- Administración de configuración de versiones
- Selección de estándares de codificación
- Selección de un esquema de integridad del software
- Revisión de la relación entre diseño y los requerimientos
- Evaluación de conformidad de la implementación respecto al diseño
- Evaluación de conformidad de la implementación respecto a los requerimientos
- Evaluación de los ambientes de desarrollo para el cumplimiento de estándares
- Evaluación de los ambientes para la realización de test unitarios, de integración y aceptación definidos
- Evaluación de las actividades del proyecto respecto a los requerimientos

1.18. Suposiciones y dependencias

La segmentación y clasificación por equipo de los jugadores de un fútbol solo es valido para vídeos de fútbol donde la cancha sea de color verde, para que el sistema trabaje de forma correcta los equipos deben tener uniformes de un color que no sea verde, además cada equipo de utilizar un color diferente.

En caso de que el arbitro salga en alguna toma y utilice un uniforme de color similar a alguno de los equipos hay gran probabilidad de sea tomado como parte de uno de los equipos.

El sistema no va a trabajar de igual forma en todas las máquinas que se instalen, el comportamiento puede variar dependiendo de la capacidad de la computadora.

2. Estándar de codificación

El proyecto está siendo desarrollado en Java, con un front-end en Angular (Typescript), por lo tanto, es necesario determinar los estándares para estos dos lenguajes.

2.1. Java

Para el desarrollo de la aplicación en Java, se escogió la guía de estilo de Google [1]. Se decidió utilizar este estándar por varias razones. Primero, por la popularidad del estilo, de las guías de estilo en la página web de GitHub este es el estándar más popular con más de 5700 estrellas y 1500 forks [2]. Segundo, el repositorio de Google incluye un archivo .xml con la especificación del estándar, el cual permite agregar el estilo al IDE de trabajo (Eclipse). Finalmente, el estándar es el más cercana al estilo ya utilizado por el grupo previamente.

2.1.1. Propiedades destacadas del estándar:

1. Estructura de un archivo fuente:
 - Licencia
 - Declaración de Package
 - Imports
 - Una única clase
2. Todo bloque es definido entre corchetes { }, inclusive los “if” de una línea, además, los bloques usan el estilo de Kernighan y Ritchie, donde:
 - No se permite una nueva línea antes de abrir el bloque.
 - Debe haber una nueva línea después de abrir el bloque.
 - Deber haber una nueva línea antes de cerrar al bloque.
 - Debe haber una nueva línea después de cerrar el bloque, solo el } sirve para cerrar un método, constructor o clase.
3. Una declaración por línea.
4. Columna con un límite de 100 caracteres.
5. Indentación será de 4 espacios.
6. Alineación horizontal no es requerida.
7. Los arreglos pueden ser declarados sobre múltiples líneas
8. Los nombres de clases usan el “UpperCamelCase”
9. Los nombres de métodos usan el “LowerCamelCase”

10. Constantes usan el “CONSTANT_CASE”
11. Toda clase pública y protegida junto con los miembros de dichas clases llevan JavaDoc, al menos que sea un método que sobrescriba al método padre, o si el método es muy simple como en el caso de “getFoo”.

2.2. Angular 2.0 (Typescript)

Debido a que se está utilizando la version más nueva de angular (Angular 2.0), no tiene sentido aplicar una guía de estilo de Javascript, ahora se trabaja con Typescript por defecto, por lo cual, se utilizará una guía de estilo para Angular 2.0 y no para el lenguaje de JavaScript en sí. La guía de estilo seleccionado es la que se encuentra en la página oficial de Angular [4]. Se seleccionó este estándar por las siguientes razones: es la guía de estilo del mismo equipo que desarrolla y mantiene Angular. Además, existe una herramienta llamada “codelyzer” que revisa el cumplimiento del estilo de forma automatizada, además existen “snippets” para el editor de Visual Code que siguen las reglas del estilo. Finalmente, este estándar no se limita solo al estilo del lenguaje en particular, también tiene reglas que se pueden aplicar a todo el programa como programa de Angular 2.0.

2.2.1. Propiedades destacadas del estándar

1. Se aplica el concepto de “Single Responsibility Principle” a todos los componentes.
2. Se busca que los nombres de archivo digan funcionalidad, después tipo y finalmente extensión, separado por punto.
3. Los nombres de símbolos usan el “UpperCamelCase”
4. Los nombres de métodos y propiedades usan el “LowerCamelCase”
5. Ponen toda la lógica de plataforma y de “bootstrapping” en un archivos main.ts
6. Los archivos de test se nombran igual a los archivos que corresponden
7. Se crea una carpeta por funcionalidad, donde el nombre de la carpeta corresponde a la funcionalidad.
8. Se utilizan servicios como “Singletons” dentro del mismo injector.

3. Diagramas

Es claro que el sistema necesita tener un desacoplamiento de la Vista (el front-end en angular) con el Modelo (el back-end que procesa el video), pues en este caso, la Vista está siendo desarrollada en un lenguaje distinto al del Modelo.

Lo anterior se logra con la utilización del patrón MVC (Model View Controller) [3]. Para este proyecto, se hará uso de este patrón, pues resuelve perfectamente el problema de comunicación que existe entre el “Frontend” y “Backend”. Este patrón se caracteriza por separar las responsabilidades de la Vista y el Modelo utilizando un Controlador que procesa todas las llamadas.

Cabe mencionar que no se agregó una capa de seguridad al diseño, ya que se planea cumplir este requerimiento sin la necesidad de implementar una pantalla de “login”. La librería de Spring, que se utiliza en la implementación del web-server, permite configurarse para sólo aceptar requests de un IP especificado, por lo cual se asignará un IP estático a las máquinas autorizadas a utilizar la aplicación.

3.1. Componentes

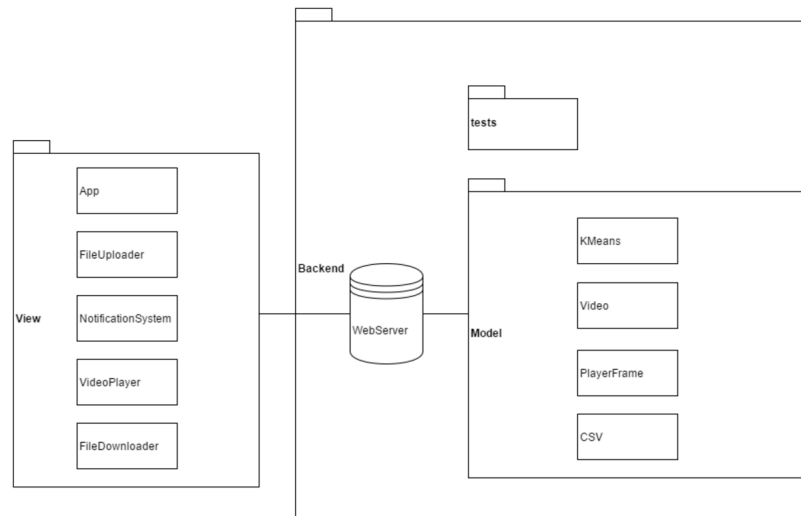


Fig 3. Diagrama de componentes del sistema Sport Analytics

3.2. UML

3.2.1. Front-End

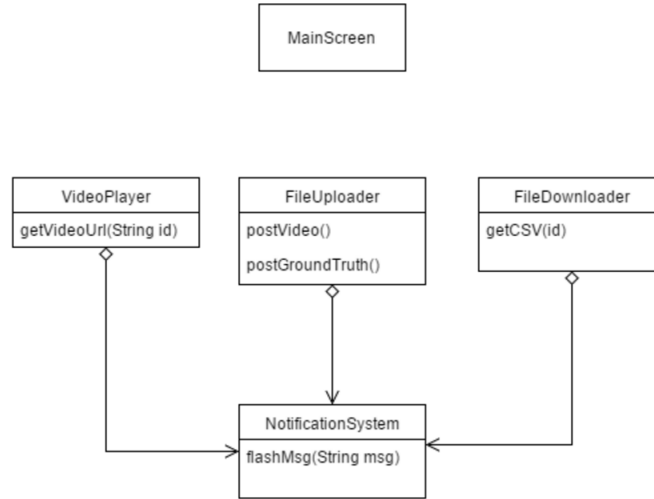


Fig 4. Diagrama UML del Front-End del sistema

3.2.2. Back-End

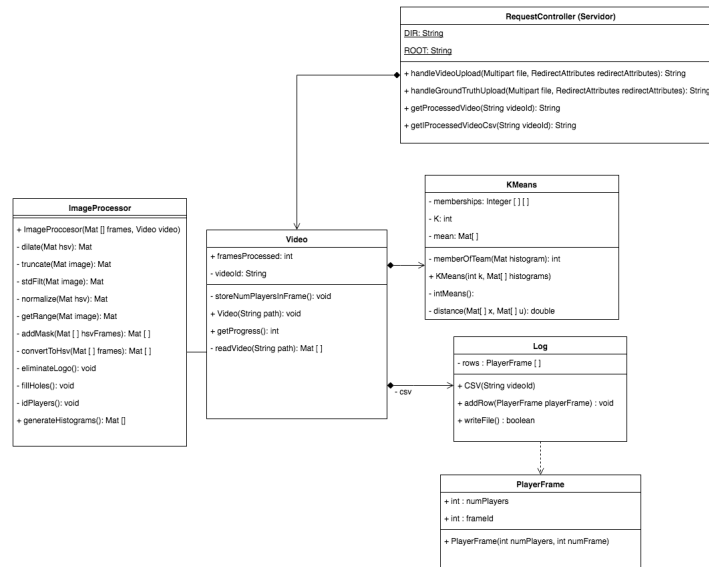


Fig 5. Diagrama UML del Back-End del sistema

4. Actividades de Aseguramiento de la Calidad del Software

4.1. Planeamiento del proceso de ACS

1. *Administración de configuración de versiones*

Tarea: Seleccionar una herramienta de control de versiones para utilizar durante el proyecto.

Responsable: Seth Stalley

Realización: Sprint 1

Posibles Herramientas: Github, GitLab, Subversion, Bazaar.

2. *Selección de estándares de codificación*

Tarea: Seleccionar estándares de codificación para el desarrollo del front-end y back-end.

Responsable: Seth Stalley

Realización: Sprint 2

Posibles Herramientas: Codelyzer, Eclipse.

3. *Selección de un esquema de integridad del software*

Tarea: Seleccionar una herramienta para la determinación de los niveles de integridad de software.

Responsable: Lucy Chaves

Realización: Sprint 3

Posibles Herramientas: IEEE Std 1012-2012 [B26].

4.2. Aseguramiento del producto

1. *Revisión de la relación entre el diseño y los requerimientos*

Tarea: Verificar que todos los requerimientos recolectados hayan sido considerados en el diseño del producto.

Responsable: Valeria Calderón

Realización: Sprint 3

2. *Evaluación de conformidad de la implementación respecto al diseño*

Tarea: Constatar que todas las clases, métodos y relaciones establecidas en el diseño, hayan sido implementadas en el código.

Responsable: Lucy Chaves

Realización: Último Sprint

3. *Evaluación de conformidad de la implementación respecto a los requerimientos*

Tarea: Constatar que todos los requerimientos funcionales se cumplan, es decir, que hayan sido implementados en el código.

Responsable: Lucy Chaves

Realización: Último Sprint

4.3. Aseguramiento del proceso

1. *Evaluación de los ambientes de desarrollo para el cumplimiento de estándares*

Tarea: Revisar que los ambientes de desarrollo seleccionados para el proyecto sean adecuados para el cumplimiento de los estándares definidos.

Responsable: Valeria Calderón

Realización: Sprint 3

2. *Evaluación de los ambientes para la realización de tests unitarios, de integración y aceptación definidos.*

Tarea: Revisar que las herramientas de prueba seleccionadas sean adecuadas para llevar acabo todos los test definidos.

Responsable: Seth Stalley

Realización: Sprint 1

3. *Evaluación de las actividades del proyecto respecto a los requerimientos*

Tarea: Constatar que las actividades definidas permiten cumplir con todos los requerimientos recolectados.

Responsable: Valeria Calderón

Realización: Sprint 3

5. Referencias

- [1] (2015). Google Java Style Guide. Retrieved August 29, 2016, from <https://google.github.io/styleguide/javaguide.html>
- [2] (2015). GitHub - google/styleguide: Style guides for Google-originated open ... Retrieved August 29, 2016, from <https://github.com/google/styleguide>
- [3] (2015). Model-View-Controller - MSDN - Microsoft. Retrieved August 29, 2016, from <https://msdn.microsoft.com/en-us/library/ff649643.aspx>
- [4] (2016). Style Guide - ts - Angular 2. Retrieved August 29, 2016, from <https://angular.io/styleguide>