Avance 1 del Proyecto Final

Jose Fernando Molina Yordan Jiménez
Leonardo Mendoza Lisa Weber

October 1, 2016

Especificación de Requerimientos del Sistema

1 Propósito del sistema

El fútbol es un deporte que mueve pasiones a nivel mundial. Es el deporte mas popular del mundo, es por ello que también es uno de los que mas dinero mueve por año, sin contar los miles de hinchas a lo largo del mundo que disfrutan del juego y esperan que si equipo gane títulos de manera regular. En el caso de un país como Argentina, el fútbol se vive con mucha mas pasión que otros países en el mundo.

El objetivo central de la elaboración de este sistema, rige en obtener datos relevantes acerca de las estrategias y los patrones de juego de un equipo en el campo de juego que podrán ser utilizados por los miembros del cuerpo técnico del Boca Juniors, donde se busa entre otras cosas detectar la posición de jugadores, tanto los de la casa como los visitantes, cuando se encuentre en proceso un partido de fútbol, o por medio de una grabación de vídeo. Así, los miembros del cuerpo informático del Boca Juniors podrán unirse posterior a la culminación de un sistema básico para la modificación y adición de nuevas funcionalidades.

2 Alcance del sistema

La elaboración del sistema de Análisis Automático de Vídeos de Fútbol consiste en realizar un programa que de manera automática, a partir de un vídeo de un partido de fútbol, detecte las posiciones relativas de los jugadores a la posición del campo de juego, para de esta manera generar datos importantes que pueden ser tomados en cuenta a la hora de planear un partido de fútbol contra un equipo en especifico. El programa debe ser lo suficientemente confiable para que los stakeholders tengan la posibilidad de utilizarlo y obtener una ventaja competitiva sobre los rivales.

El sistema por desarrollar no tiene solo que ser confiable en la generación de datos, sino que los datos deben poder ser analizados para su interpretación. Ademas, el sistema debe ser entregado al equipo de TI del equipo Boca Juniors, por lo que la mantenibilidad del mismo debe ser alta.

La entrega del sistema esta sujeta a la aprobación de la junta directiva. La entrega del producto incluye la instalación física del sistema en un servidor ya sea propio del club o externo al mismo, el código fuente, así como una capacitación para las personas que utilizaran la herramienta para minimizar el tiempo de aprendizaje y maximizar el aprovechamiento del mismo y otra capacitación para los encargados del mantenimiento del sistema para minimizar los problemas de aprendizaje del equipo de TI sobre el producto.

3 Resumen del Sistema

3.1 Contexto del sistema

El nombre de la empresa que desarrollara el software es Sport Analytics. Recientemente el actual presidente del Boca Juniors, Daniel Angelici ha contactado a la empresa Sport Analytics con interés de desarrollar una plataforma computacional automática o semi-automatizada que permita analizar diversos aspectos de desempeño del club Boca Juniors. Entre las conversaciones sostenidas con Angelici y el resto de la junta directiva (vicepresidentes: Horario Paolini y Rodolfo Ferrari, Dario Richarte, secretarios: Gustavo Ferrari, Carlos Aguas y Pedro Orgambide), se ha sustraído necesidades a resolver por el sistema potencialmente a desarrollar para el Boca Juniors.

Los usuarios del sistema serán miembros del club: El director técnico y miembros de la junta directiva. La empresa Sport Analytics se ha propuesto entregar una primera iteración del proyecto, usando diferentes estándares de calidad y herramientas para cumplirlos, para alrededor de Noviembre del presente año.Ello puesto que el equipo de informáticos del Boca Juniors tiene planteado involucrarse en el desarrollo del eventual proyecto. Según la apreciación de Angelici, el Boca Juniors accederá en continuar con el proyecto o no.

3.2 Funciones del sistema

- Obtener, a partir de un video de futbol, los segmentos que puedan ser utilizados para analizar información útil.
- Generar un informe con la duración de cada escena detectada, utilizando el tiempo de inicio y el tiempo de final de la escena como referencias.
- Obtener de manera no supervisada datos sobre la consistencia de las diferentes líneas de juego: defensiva, medio campo y ofensiva.
- Generar un informe con las posiciones de cada jugador de un equipo específico en un cuadro del vídeo.

3.3 Características de los usuarios

- **Director Técnico:** Es el principal interesado en acceder a los datos generados por el sistema. El director técnico es el usuario principal, es la persona que a partir de los datos generados por el sistema toma decisiones tácticas a ser implementadas por el equipo. Es importante que conozca todas las capacidades del sistema para sacarle el mayor provecho posible a la herramienta. Debe participar activamente en el aporte de conocimiento futbolistico que pueda ser utilizado para la mejora del sistema de analisis.
- **Junta Directiva:** Es el dueño del sistema y del negocio. El Director Técnico le responde a la Junta Directiva. Es la encargada de definir los requerimientos funcionales del sistema y de aprobar cualquier cambio en los mismos. No sera la encargada de utilizar el sistema la mayor parte del tiempo, pero es la parte mas interesada en su correcto despliegue.

4 Requerimientos Funcionales

4.1 REQ-1: Separación temporal

Descripción: Separar un vídeo de fútbol en escenas validas para realizar un análisis de juego. Por escenas validas se entiende aquellas escenas donde el campo de juego sea el objeto predominante en la imagen.

Prioridad: Media

4.2 REQ-2: Informe sobre separación temporal

Descripción: Generar un informe con la duración de cada escena detectada, utilizando el tiempo de inicio y el tiempo final de la escena como referencias. El informe debe ser escrito en formato XML y cada escena detectada debe incluir: Tiempo de inicio, tiempo final, cantidad de cuadros.

Prioridad: Media

4.3 REQ-3: Consistencia de las lineas de juego

Descripción: Obtener de manera no supervisada datos sobre la consistencia de las diferentes líneas de juego: defensiva, medio campo y ofensiva en un intervalo de tiempo. Para la consistencia de cada linea de juego se divide el campo en 3 sectores: uno por cada linea de juego. Luego, se realiza una regresión lineal de la posicion de los jugadores respecto al eje x, se toma la distancia entre cada jugador y la función de regresión y se divide entre el total de jugadores tomados en cuenta. Una linea optima seria un valor de 0. El indice estara normalizado respecto al tamaño del sector analizado.

Prioridad: Alta

4.4 REQ-4: Informe sobre consistencia de lineas de juego

Descripción: Generar un informe con la consistencia de las lineas de juego en el intervalo de tiempo especificado en **REQ-3**. El informe debe contener el indice de consistencia de cada linea, en formato XML.

Prioridad: Alta

4.5 REQ-5: Informe sobre posiciones de los jugadores.

Descripción: Generar un informe con la posición de los jugadores de un equipo en un cuadro especifico. Para describir la posición se considera una escala desde -100 hasta 100 en dimensión X y de 50 hasta -50 en dimensión Y, tomando como punto (0,0) el punto central de la cancha.

Prioridad: Alta

4.6 REQ-6: Cargar un vídeo.

Descripción: El sistema podrá cargar un vídeo en memoria proporcionado por el usuario el cual se espera que se sea analizado.

Prioridad: Alta

4.7 REQ-7: Descargar un vídeo como resultado del análisis.

Descripción: El sistema ejecutara las características necesarias para lograr obtener un vídeo resultado con la ubicación de los jugadores en el campo de juego por medio de umbrales, que sera enviado desde el servidor al front-end para poder ser descargado por el usuario final.

Prioridad: Alta

4.8 REQ-8: Informe de los jugadores de cada equipo detectados en el campo de juego.

Descripción: Generar un archivo csv, con información de la cantidad de jugadores por equipo detectados en el campo de juego por cuadro dentro del vídeo solicitado.

Prioridad: Alta

4.9 REQ-9: Numero de jugadores detectados.

Descripción: El vídeo resultado brindado al usuario final, indicara en pantalla la cantidad de jugadores se encuentran detectados en el cuadro que se este presenciando, dicho resultado se ubicara en la esquina superior izquierda.

Prioridad: Alta

4.10 REQ-10: Regiones correspondientes a los jugadores.

Descripción: Dentro de la composición del vídeo resultado, el sistema asignará los campos correspondientes o blobs de los jugadores detectados por el algoritmo, superponiendo los umbrales generados con el algoritmo de outsu al vídeo origen.

Prioridad: Alta

4.11 REQ-11: Etiquetas por equipo en los jugadores detectados.

Descripción: El sistema analizará las regiones de los jugadores visibles por medio de un análisis de sus píxeles que abarca el algoritmo k-medias. Dicho proceso brindará como resultado la pertenencia de cada jugador a un equipo, con ello se etiquetarán a los jugadores en el vídeo resultado con su respectivo equipo.

Prioridad: Alta

4.12 REQ-12: Tiempo en procesar.

Descripción: El sistema presentará el tiempo de procesar el análisis del vídeo al usuario luego de terminar el proceso.

Prioridad: Alta

4.13 REQ-6: Exactitud en los resultados.

Descripción: El sistema aceptará hacer revisiones contra un archivo ground-truth para medir la precisión de su ejecución.

Prioridad: Alta

5 Requerimientos de usabilidad

5.1 UREQ-1: Detección de Jugadores

Descripción: El sistema deberá alcanzar un índice de 0.8, durante los 100 frames estipulados dentro de los videos, a la hora de clasificar y obtener los jugadores que se encuentran en el terreno de juego.

Descripción de la metrica: Tomar 100 frames válidas (toma televisiva del campo de juego) del video por analizar y comparar la cantidad de jugadores detectados contra un archivo de ground-truth con la cantidad verdadera de jugadores que se encuentran en el frame.

Fórmula: $\sum_{n=1}^{100} \frac{A_n}{B_n}$ donde
A_n = Jugadores detectados en el cuadro N
B_n= Cantidad de jugadores en el cuadro N

Prioridad: Baja

5.2 UREQ-2: Completitud de la descripcion

Descripción: La documentación de los requerimientos funcionales, destinado a usuarios finales ya sea manuales, tutorial, etc, deberán tener un promedio de completitud de 95%.

Descripción de la métrica: Comparar la lista de requerimientos funcionales contra los manuales de usuario, las paginas de ayuda y tutoriales realizados. Cuántas de las funcionalidades vienen descritas en alguno de estos artefactos? Tomar la cantidad de funcionalidades que vienen descritas en los artefactos y dividirlos entre la cantidad de requerimientos.

Fórmula: $\frac{A}{B}$ donde
A = Cantidad de reqeurimientos funcionales explicados
B = Total de requerimientos funcionales

Prioridad: Media

5.3 UREQ-3: Facilidad de aprendizaje

Descripción: Los usuarios que utilice el software responderán con una curva de aprendizaje promedio menor a 5 minutos.

Descripción de la métrica: Tomar un usuario que nunca haya utilizado el sistema. Pedirle que realice una serie de actividades funcionales con el software y medir el tiempo que le toma hacerlo. Luego, dividir entre la cantidad de funciones para obtener el tiempo promedio que le toma.

Fórmula: $\frac{A}{B}$ donde
A = Tiempo total tomado
B = Cantidad de actividades.

Prioridad: Baja

5.4 UREQ-4: Atractividad de la interfaz gráfica

Descripción: Los usuarios finales clasificarán la interfaz gráfica respondiendo de la mejor manera esperable, obteniendo como resultado una percepción positiva mayor al 80% promediado entre todas las respuestas de los usuarios.

Descripción de la métrica: Mediante un cuestionario con rubricas sobre la interfaz gráfica y una calificación entre 1 y 10, encuestar a 10 posibles usuarios del sistema acerca de su percepción de la interfaz gráfica.

Fórmula: $\frac{A}{B}$ donde
A = Suma de las clasificaciones de cada rúbrica
B = Puntaje máximo.

Prioridad: Alta

5.5 UREQ-5: Disponibilidad

Descripción:EL sistema debe poseer una estructura fácil de modificar, por esto el sistema a un nivel superior del 90% se podrán reparar su estructura en un tiempo óptimo.

Descripción de la métrica: Probar el producto bajo un funcionamiento normal. Monitorear el tiempo que trabaja sin problemas y, en caso de fallo, el tiempo que se toma en volverlo a poner a funcionar.

Fórmula: $\frac{T_o}{T_o+T_r}$ donde
T_o = Tiempo operativo
T_r = Tiempo en reparaciones

Prioridad: Media

5.6 UREQ-6: Complejidad de mantenimiento

Descripción: La eficiencia en la estructura del código de la lógica de negocios contendrá un índice menor al 20% de complejidad, especialmente para evitar futuros contratiempos a la hora de realizar modificaciones por un usuario nuevo que se integre a la construcción o mantenimiento.

Descripción de la métrica: Analizar el comportamiento del usuario externo que se encuentra realizando cambios en alguna clase, y observar cuando presenta mayor dificultad a la hora de enfrentar algún cambio.

Fórmula: $\frac{A}{B}$ donde
A = Cambios que presentaron dificultad de cambio
B = Cantidad total de cambios.

Prioridad: Media

5.7 UREQ-7: Capacidad de diagnóstico de errores

Descripción: El uso de buenas prácticas de programación es indispensable para cumplir este objetivo, con el fin de evaluar y obtener un promedio de dificultad de obtener la precedencia de un error menor a 20% que se realizarán durante la etapa de pruebas.

Descripción de la métrica: Mientras el software se encuentra en etapa de pruebas, se observa la complejidad que presenta el usuario al analizar el origen del problema o error detectado.

Fórmula: $\frac{A}{B}$ donde
A = Errores difíciles de detectar
B = Cantidad total de errores

Prioridad: Media

5.8 UREQ-8: Adaptación del sistema a navegadores web

Descripción: El Boca Juniors brindará datos acerca de los navegadores web que se utilizen, con ello ejecutar pruebas y esperar que exista un porcentaje de adaptibilidad a los distintos navegadores mayor al 85%.

Descripción de la métrica:

Fórmula: $\frac{A}{B}$ donde
A = Numero de navegadores soportados
B = Cantidad total de navegadores probados

Prioridad: Baja

5.9 UREQ-9: Tiempo en suceder colapsos

Descripción: El sistema deberá adaptarse a un ambiente de funcionamiento con distintas páginas web trabajando al mismo tiempo, por ello es necesario que el indice de fallas que puedan ocurrir por dicha necesidad sea menor al 25%.

Descripción de la métrica: Ejecutar pruebas del sistema ejecutándose concurrentemente con otros sistemas, y medir el tiempo que falla la pagina o hay pérdida significativa de eficiencia en nuestro sistema en un tiempo total.

Fórmula: $\frac{T_a}{T_o}$ donde
T_a = Tiempo con comportamiento anormal
T_o = Tiempo operativo.

Prioridad: Baja

6 Requerimientos de eficiencia

6.1 EREQ-1: Auditabilidad de acceso

Descripción: La sección de seguridad de los usuarios, tendrá un promedio de 95% de eficiencia a la hora de acceder a los datos de bitácora almacenados en la ejecución de los inicios de sesión del sistema.

Descripción de la métrica: Realizar login 50 veces bajo diferentes usuarios. Probar con cada usuario un acceso con la contraseña correcta y otro con la contraseña incorrecta. Por cada intento de acceso, el sistema debe escribir en la bitácora el nombre de usuario que intentó hacer login y si este fue exitoso o no.

Fórmula: $\frac{A}{100}$ donde
A=Cantidad de veces que el login fue exitoso

Prioridad: Media

6.2 EREQ-2: Integración de los Módulos

Descripción: Durante la integración de los módulos importantes del sistema, su porcentaje de aprobación de las pruebas construidas deberán ser superiores a un 80%, con el fin de que el tiempo destinado a este proceso sea lo mayor reducible posible.

Descripción de la métrica: Correr todas las pruebas de integración definidas en el plan de pruebas. Dividir la cantidad de pruebas que fueron exitosas entre el total de pruebas realizadas.

Fórmula: $\frac{A}{B}$ donde
A = Pruebas de integracion exitosas
B = Cantidad de pruebas de integracion realizadas.

Prioridad: Media

6.3 EREQ-3: Grado de manejo de excepciones

Descripción: Las secciones necesarias dentro de el sistema, podrá a un nivel mayor al 75% recuperarse y afrontar las excepciones generadas en tiempo de ejecución.

Descripción de la métrica: Por cada invocación de un método, revisar todas las posibles excepciones que pueden generarse de la llamada y asegurarse que todas hayan sido manejadas por un bloque try-catch.

Fórmula: $\frac{A}{B}$ donde
A = Cantidad de excepciones manejadas
B = Cantidad de excepciones que pueden ocurrir

Prioridad: Alta

6.4 EREQ-4: Detección de fallas

Descripción: El total de pruebas realizadas al sistema final, tendrá un porcentaje de aprobación de 9 aceptadas de 10 ejecutas con el fin de reducir tiempo en reparación de errores.

Descripción de la métrica: Realizar todas las pruebas unitarias y de integración definidas en el plan de pruebas. Tomar la cantidad de pruebas que fueron exitosas y dividirlas entre todas las pruebas realizadas.

Fórmula: $\frac{A}{B}$ donde
A = Cantidad de pruebas exitosas
B = Cantidad de pruebas realizadas

Prioridad: Baja

6.5 EREQ-5: Píxeles analizados por segundo

Descripción: La velocidad de ejecución del análisis de píxeles debe ser la óptima, por ello el software construido ejecutará un monto mayor a 24000000 píxeles por minuto.

Descripción de la métrica: Ejecutar 50 pruebas de análisis de videos donde se cuantifique término de tiempos y píxeles cuántos pueden ser ejecutados por minuto.

Fórmula: $\frac{\sum_{1}^{100} \frac{H*W*F}{T_n}}{100}$ donde
H = Alto del vídeo en píxeles
W= Ancho del vídeo en píxeles
F = Frames por segundo
T_n=Tiempo en segundos que tardo el análisis del frame N.

Prioridad: Baja

6.6 EREQ-6: Utilización de recursos

Descripción: Los recursos deben ser un factor importante por controlar, por ello durante las pruebas se medirá la cantidad de Bytes mayor utilizados, para lograr que el sistema resultado utilice un porcentaje menor a 80% de la memoria disponible.

Descripción de la métrica: Ejecutar 100 veces el análisis de videos, de los cuales se almacenará el número de bytes mayor utilizado en cada prueba. Luego se obtiene un promedio.

Fórmula: $\frac{\sum_{n=1}^{100} A_n}{100}$ donde
A_n = Cantidad maxima de memoria utilizada en la ejecución N

Prioridad: Media

6.7 EREQ-7: Reinicio en pruebas

Descripción: La calidad aplicada en la construcción de las pruebas deben poseer el mejor punto de entendimiento para el auditor, esto para lograr que consulte el código fuente en un promedio menor al 15% del total pruebas.

Descripción de la métrica: Mientras se realizan pruebas externas observar el comportamiento del ejecutante cuando debe reiniciar una prueba, e ir a revisar el código fuente para observar detalladamente lo que sucede.

Fórmula: $\frac{A}{B}$ donde
A = Numero de pruebas cuando se reinicio
B = Cantidad total de pruebas

Prioridad: Media

6.8 EREQ-8: Funciones reemplazadas

Descripción: El equipo contabilizará las funciones que reemplaza el sistema actual con las cuáles se encontraban en la entidad, para que el 90% sean sustituidas por completo.

Descripción de la métrica: Contar las funciones que se sustituyen en la entidad comparados con el total de funcionalidades que existen antes en el equipo, esto por medio de encuestas a los futuros usuarios.

Fórmula: $\frac{A}{B}$ donde
A = Numero de métodos reemplazadas
B = Cantidad total de métodos en el modulo.

Prioridad: Alta

7 Interfaces del sistema

7.1 Interfaz para elemento humano

El sistema de interfaces gráficos esta compuesto por una aplicación web en Angular2. Dicha aplicación podrá ser accesada solamente desde la red interna del equipos Boca Juniors. Esta interfaz gráfica se encarga de subir archivos de vídeo, mostrar el progreso al usuario (tanto de subida, como de trasformación de vídeo) y luego devuelve el archivo de vídeo resultante, con el respectivo análisis realizado. De la misma manera, por medio de la interfaz gráfica

7.2 Interfaz Java

El verdadero análisis del vídeo sera realizado por un archivo desarrollada con el lenguaje de programación Java junto con la librería OpenCV para el manejo de análisis gráfico por computadora. El programa que realice el análisis recibirá por parámetro la ruta del video que debe ser analizado y lo modificara sin realizar una copia del vídeo original. Es importante que esta interfaz tenga la capacidad de notificar a la interfaz en NodeJS acerca de su progreso de manera periodica.

7.3 Interfaz NodeJS

La puerta de entrada al back-end del Analizador de Vídeos de Fútbol sera el servidor escrito en el lenguaje NodeJS. Este servidor se encargara de autenticar a los usuarios que quieran hacer uso de la herramienta para evitar accesos no autorizados. Este servidor sera el encargado de recibir por parte de AngularJS el video que debe ser analizado, subir el video a servidor y ejecutar el Analizador en Java, reportando progreso periódicamente al Front-End de AngularJS.

7.4 Interfaz sistema Boca Juniors

El sistema por realizar debe ser integrable al sistema ya existente del equipo Boca Juniors. Es importante la constante comunicación entre el equipo de desarrollo y los ingenieros del equipo para lograr una integración exitosa de los módulos, haciendo vital importancia a la autenticacion de las personas autorizadas a utilizar el sistema y el almacenamiento seguro de los datos generados.

8 Operaciones del sistema

8.1 Requerimientos de integración de sistema del personal

Para la elaboración del proyecto se requiere concentrar la mayor cantidad de fuerza de trabajo del personal en el área de investigación y de implemetación de prototipos de "procesamiento de vídeo". Esto debido a que esta área es una de las mayores prioridades en el proyecto y con la que se cuenta menos información para ser desarrollada.

8.2 Mantenibilidad

8.2.1 Tiempo

Se espera tener un servició que este disponible las 24 horas del día. En caso de que sea necesario realizar mantenimiento sobre algún componente, este deberá hacerse durante horas no laborales del club, preferiblemente entre las 00:00 y las 4:00 hora local. El mantenimiento fisico del sistema sera responsabilidad del club Boca Juniors, siendo Sport Analytics responsable solamente de actualizaciones al sistema o reparación de fallas no relacionadas con los sistemas físicos.

8.2.2 Frecuencia de mantenimiento

Semanalmente habrá un encargado de revisar el buen funcionamiento del sistema, esto lo hará por medio de la revisión de archivos log, con información sobre los resultados obtenidos durante la semana.

8.2.3 Tiempo de reacción

La cantidad de tiempo que pase entre la caída del sistema y el primer contacto con el personal encargado de mantenimiento deberá ser menor a 30 minutos. El tiempo esperado para recobrar su funcionamiento normal deberá ser menor a 12h.

8.3 Confiabilidad

8.3.1 Completar proceso de subida-análisis y descarga de vídeo

Al menos 99 de cada 100 vídeos procesados deberán cumplir exitosamente con la obtención de resultados esperado, esto es, comportarse de acuerdo a los parámetros aceptados por las métricas expuestas en apartados anteriores.

8.3.2 Precisión de posición de jugadores

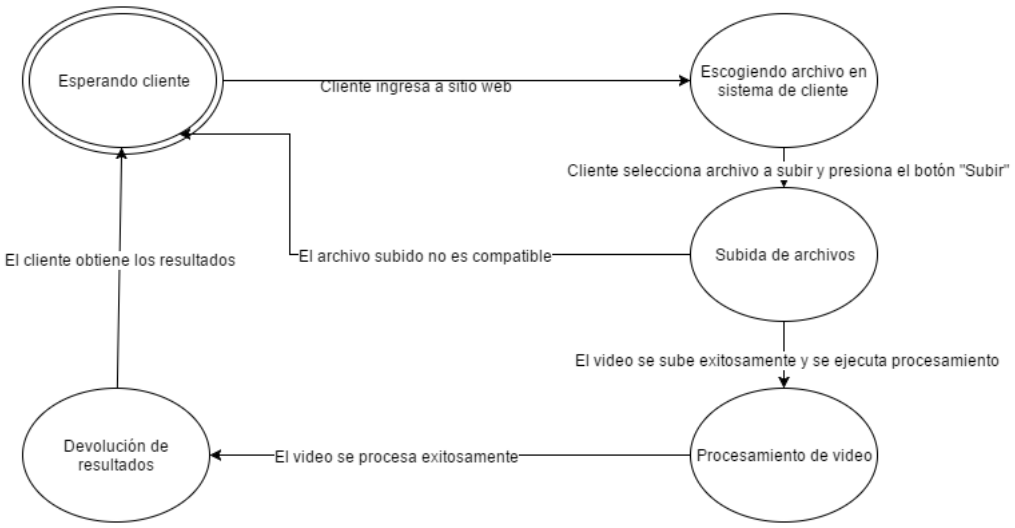
El máximo fallo permitido es de 4 puntos sobre los ejes Y y X.En la escala definida que va de -50 a 50 en ambos ejes.

8.3.3 Jugadores detectados

800 de cada 1000 jugadores a los que se les detecte equipo deberán ser detectados correctamente.

9 Modos y estados del sistema

A continuación el diagrama de estados que ilustra los posibles estados en los cuales se podría encontrar el sistema en un determinado momento



10 Caracteristicas Fisicas

10.1 Requerimientos fisicos

El sistema para su estructura necesita de un servidor corriendo las 24 horas de los 7 días disponibles, por ello la estructura que aloje este dispositivo, debe poseer un espacio o cuarto considerable con medidas mínimas de 5 metros de largo por 3 metros de ancho. Esto además este sector asignado al servidor debe ser completamente cerrado con una única entrada y salida. Es necesario que el servidor lleve su respectivo escritorio, silla ergonómica, teclado, mouse, monitor y sistema de batería para afrontar eventuales cortes de energía, cada uno etiquetados con su respectivo número de identificación de activo. La entrada de iluminación debe ser únicamente artificial por medio de lámparas.

10.2 Requerimientos de adaptabilidad

El servidor deberá poseer como mínimo dos entradas de memoria ram para evitar futuros problemas en este dispositivo y poseer estos de respaldos, o para una futuro aumento de capacidad debido al crecimiento del sistema o usuarios. Además es posible contener dos tarjetas de red en un futuro por ello debe existir un Jacks de un conector RJ-45 libre además del existente.

10.3 Requerimientos Ambientales

La temperatura del cuarto del servidor debe ser regulada, por medio de un sistema de aire acondicionado que mantenga la temperatura en una media de 15 grados centigrados. Además como se explico anteriormente, el cuarto debe ser cerrado, por ello cualquier tipo de factor ambiental ya sea humedad, luz solar, lluvia, plantas o animales debe presentar en lo mínimo posible. El sector debe ser fumigado al menos una vez cada 6 meses para evitar la existencia de plagas de insectos o cualquier otro. Otro requisito importante, el edificio debe

contener un pararrayos que mitigue cualquier problema que ocurre por tormenta eléctrica y afecte directamente a los bienes de la entidad contenidos.

10.4 Seguridad del Sistema

El edificio de donde se encuentre ubicado el servidor debe poseer un sistema de vigilancia por cámara de 24 horas los 7 días de la semana, además es indispensable de personal de seguridad que se encargue de velar por la integridad de los bienes como de restringir el acceso al personal autorizado para uso o mantenimiento del servidor. Ademas este personal mantendrá un sistema de bitácora para ver los accesos y salidas del cuarto.

10.5 Manejo de la información

La información controlada por el servidor debe ser integra y manejada con la mayor seguridad pueda posible, la capacidad de almacenamiento del dispositivo central será de 10 TB, con lo cual aseguramos que la falta de espacio en un largo plazo sea poco posible. Además el servidor almacenará los backups diarios en un host remoto contratado por la entidad para mantener la integridad de datos del dispositivo en caso de un altercado.

10.6 Regulaciones y políticas

Las políticas de uso de la plataforma del sistema respeta y no brinda ningún trato fuera de mostrar y almacenar los datos ingresados por el usuario, evitando por completo el uso de la información para la minería de datos. El departamento legal de la entidad se encargará de esta sección y elaborará los términos y condiciones de uso del sistema donde se abarque que los datos de los usuarios son protegidos. Además el sistema será regulado por los derechos de autor de la entidad del Boca Juniors, por lo cual si existe un uso no permitido del sistema el departamento legal accederá a las leyes protectoras de los derechos de uso con quien use el sistema sin permiso alguno.

10.7 Mantenimiento del ciclo de vida del sistema

El departamento de mantenimiento de la entidad se encargará de velar, por el estado físico de los dispositivos utilizados, donde periódicamente se revisaran los dispositivos para evitar daños en el hardware inesperado.Además se contendrán los manuales de usuario en caso de sufrir problemas con el servidor, estos mismos contendrán pasos básicos para afrontar una situación mientras los técnicos encargados no se encuentren.

10.8 Empaque, manejo, envió y transporte

El personal encargado de la entidad, manejará con estándares de seguridad el transporte de de cualquier dispositivo que incluya cuidado para ello, solo el personal capacitado y con permiso podrá acceder a este tipo de proceso.

10.9 Verificación

Vease la seccion 5 y 6 de este documento.

10.10 Asunciones y Dependencias

10.10.1 Sistema operativo

Es necesario que el servidor se encuentre con el sistema operativo instalado el cual deberá ser Microsoft Windows Server 2014 con los distintos drivers correspondientes al hardware del servidor.

10.10.2 Red institucional

En la entidad del Boca Juniors, se asume la existencia una red dentro de la institución, con la capacidad de ingresar internet y compartir archivos. Esto para logar poner en funcionamiento el sistema dentro de la institución.

10.10.3 Departamento de Mantenimiento

El encargado de controlar el ciclo de vida de los dispositivos electrónicos que forman parte del sistema debe ser brindado o contratado por la institución del Boca Juniors.

10.10.4 Estructuras de hospedaje del hardware

El cuarto de servidor especificado se espera que sea proporcionado por la entidad Boca Juniors., con las características deseadas.

10.10.5 Recursos

El recurso del hardware especificado, la electricidad o humano (Técnicos, Oficiales de seguridad, etc) deben ser contratados por la entidad Boca Juniors.

Estándar de código escogido

1 Estandar de codigo para Java

Para implementar el código de Java se utilizará el estándar de "Google Java Style". Las razones por las que se va a implementar este estadar son:

- Amplia cobertura de diferentes áreas relevantes durante el desarrollo de codigo
- Google posee una muy alta respetabilidad en el ámbito del desarrollo de software
- Es uno de los estandares más utilizados a nivel mundial
- Este estándar está diseñado para ser fácilmente verificable.

El estandar es accesible a traves del siguiente enlace: `https://google.github.io/styleguide/javaguide.html`

1.1 Herramientas para verificación de estándar en Java

Para la verificación de la aplicación correcta de los estandares se utilizará "Checkstyle".Las razones son:

- Esta herramienta cuenta con soporte para "Google Java Style"
- Esta herramienta es soportada por medio del GUI de desarrollo Eclipse

1.2 Algunos ejemplos sobre su especificación:

1.2.1 Cambio de linea despues de brackets

```
return () -> {
  while (condition()) {
    method();
  }
};

return new MyClass() {
  @Override public void method() {
    if (condition()) {
      try {
        something();
      } catch (ProblemException e) {
        recover();
      }
    } else if (otherCondition()) {
      somethingElse();
    } else {
      lastThing();
    }
  }
};
```

1.2.2 Definición de arreglos

```
new int[] {
  0, 1, 2, 3
}

new int[] {
  0, 1,
  2, 3
}
```

```
new int[] {
  0,
  1,
  2,
  3,
}

new int[]
    {0, 1, 2, 3}
```

1.2.3 Camel Case

Correct	Incorrect
`XmlHttpRequest`	`XMLHTTPRequest`
`newCustomerId`	`newCustomerID`
`innerStopwatch`	`innerStopWatch`
`supportsIpv6OnIos`	`supportsIPv6OnIOS`
`YouTubeImporter` `YoutubeImporter` *	

2 Estandar de codigo JavaScript

Para este caso se usará el estándar "Jquery JavaScript Style Guide". Las razones:

- Este estándar se compone por una serie de reglas muy concretas y fáciles de verificar
- Jquery es una de las plataformas más respetadas para JavaScript

El estandar es accesible a traves del siguiente enlace: `https://contribute.jquery.org/style-guide/js/`

2.1 Herramientas para verificación de estándar en JavaScript

Para verificar el código se utilizará "jscodesniffer". Las razones:

- Se puede hacer una verificación agil y rapida con solo subir el codigo a la pagina

2.2 Algunos ejemplos sobre su especificación:

2.2.1 Malas practicas

```
1 // Bad
2 if(condition) doSomething();
3 while(!condition) iterating++;
4 for(var i=0;i<100;i++) object[array[i]] = someFn(i);
```

2.2.2 Buenas practicas

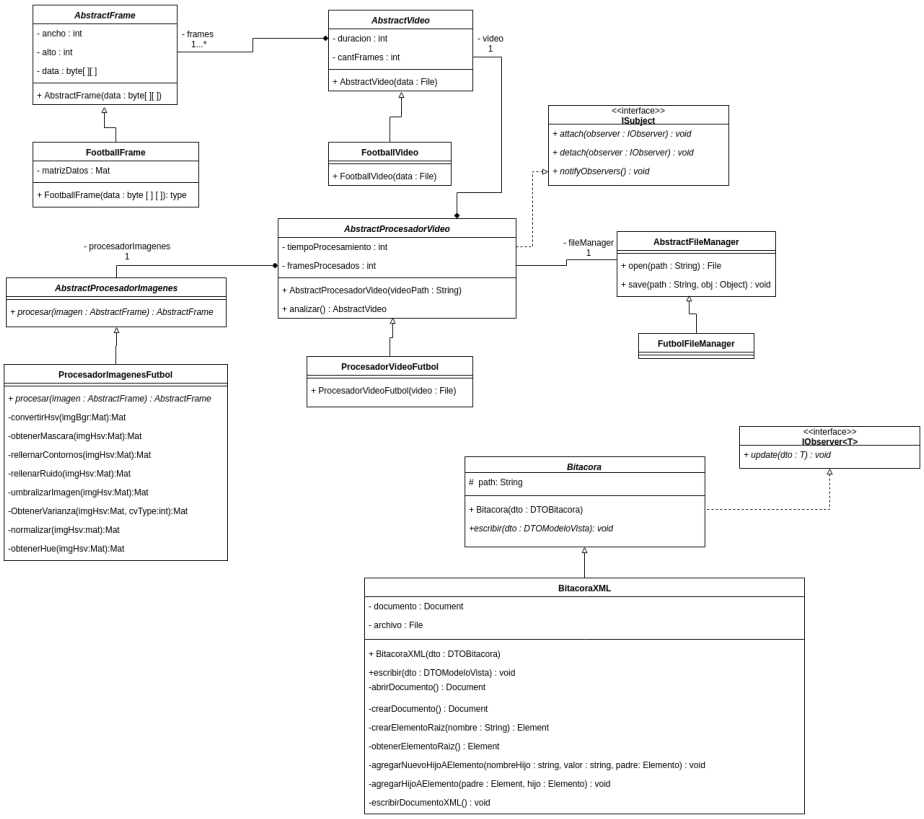
```
var i = 0;

if ( condition ) {
    doSomething();
} else if ( otherCondition ) {
    somethingElse();
} else {
    otherThing();
}

while ( !condition ) {
    iterating++;
}

for ( ; i < 100; i++ ) {
    object[ array[ i ] ] = someFn( i );
}

```

3 Conclusión

Para este proyecto se trabajará con Google Java Style y Jquery JavaScript Style Guide. Ambos estandares son ampliamente utilizados en el mercado por lo que no solo permitirán generar un código claro para el proyecto si no que permitirán al equipo conocer y entender mejor los métodos más utilizados para generar la estructura del código. Con esto se espera perfeccionar esta técnica durante el desarrollo de este producto y tambien para el desarrollo de productos posteriores.

Diagramas

1 Diagrama de Clases



1.1 Patrones implementados

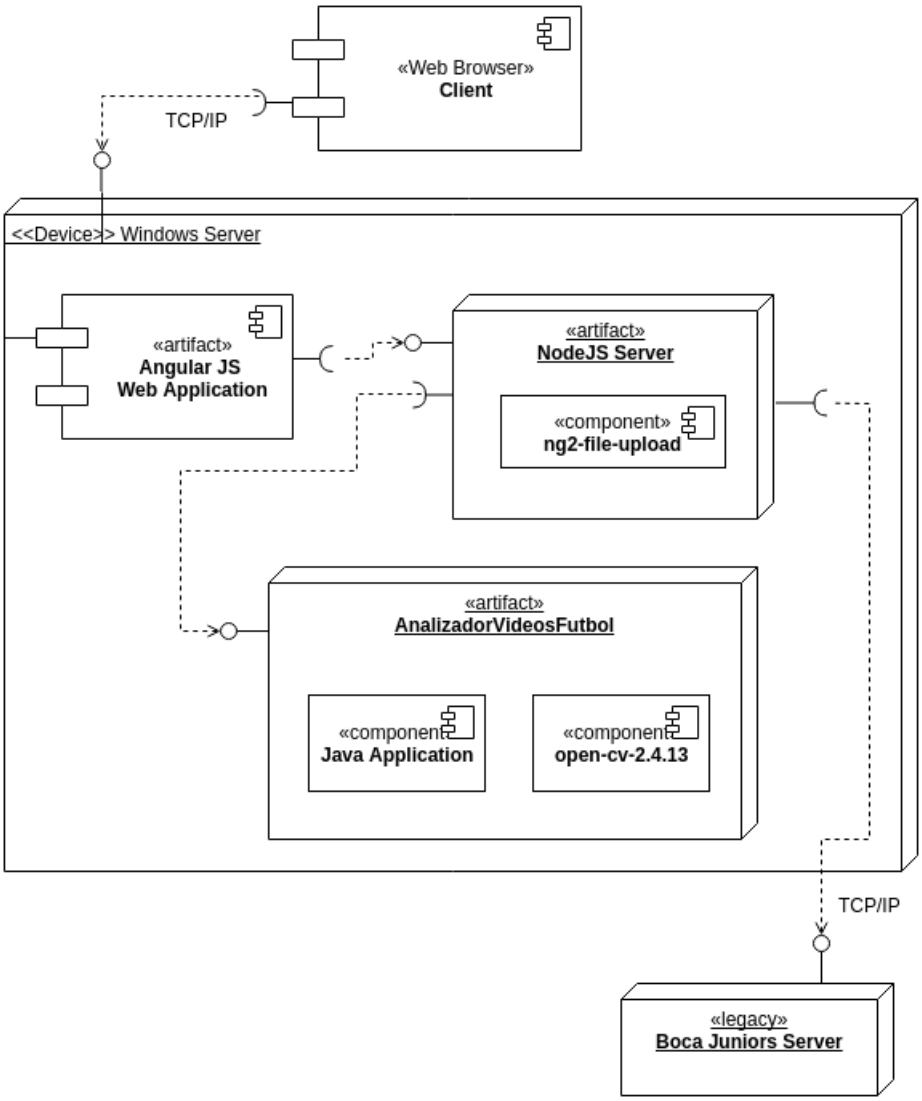
1.1.1 Observer

El patrón Observer es implementado entre la clase Bitácora (encargada de escribir la bitácora a disco) y la clase AbstractProcesadorVideo, clase encargada de realizar el procesamiento del vídeo, para que cada cierto tiempo el procesador del vídeo notifique a la bitácora acerca del avance del procesamiento y lo que debe ser escrito en la bitácora.

1.1.2 Strategy

Siguiendo los principios de diseño SOLID, el diseño toma en cuenta la posible extensión de mas funcionalidades al sistema por medio del patron Strategy mediante las clases abstractas AbstractProcesadorVideo, AbstractVideo, AstractFrame... Cada una de estas clase abstrae la funcionalidad que debe ser implementada, pero es responsabilidad de las clases concretas reaizar la implementacion de los algoritmos o metodos declarados en las clases abstractas, de esta manera es sencillo extender el análisis de otro tipo de vídeos con diferentes algoritmos, técnicas o librerías.

2 Diagrama de Componentes



3 Bibliografia

References

[1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, Massachusetts, 1st edition, 1995.

Actividades de Aseguramiento de la Calidad de Software

Durante el transcurso de este documento, se establecerán actividades a seguir para cumplimientos de estándares de calidad en las distintas secciones, en el proceso o directamente en el producto final, que abarca la construcción de este proyecto. Dichas actividades serán basadas en estándares especificados en el IEEE 730-2014.

A continuación empezaremos a especificar cada actividad que se realizarán:

1 Cumplimiento del diagrama de clases

1.1 Descripción de la actividad

En vista a que la arquitectura de la lógica de negocios es una sección delicada, se debe asesorar que la codificación principal del software siga las reglas establecidas el diseño de clases UML definido. Para ello, se revisara que la codificación responda al diseño de UML en todo momento del proyecto.

1.2 Propósito

Es necesario que exista una congruencia total entre el diseño y el código implementado, por motivos de mantenimiento y adicción de nuevas funcionalidades, con el fin de poseer una estructura definida que cualquier usuario con conocimientos de software pueda entender y facilitar el entendimiento del sistema para futuras adiciones o modificaciones.

1.3 Valida para la iteración o sprint

Se realizará esta actividad luego cada iteración.

1.4 Actividades

1. Enumerar los cambios realizados en el código fuente de la aplicación y asegurarse de modificar el diseño de manera acorde.
2. Si existen inconsistencias entre el código y diseño, enumerar dichas incongruencias y modificar el código o diseño para reflejar el estado actual del proyecto.
3. Actualizar la bitácora de cambios con las modificaciones realizadas.

1.5 Encargado

Fernando Molina

1.6 Herramienta de ayuda

EclipseUML Free Edition

2 Cumplimiento del estándar de código

2.1 Descripción de la actividad

El estándar de código que se ha establecido debe ser llevada siguiendo todas sus distintas reglas. Para ello es necesario corroborar que las tareas realizas sigan dichos lineamientos, para que el código después de su entrega sea lo más claro posible. Se revisará que el código de la aplicación siga el estándar escogido.

2.2 Propósito

Verificar que el estándar de código escogido sea cumplido de manera absoluta, para de esta manera facilitar la facilidad de lectura y entendimiento del codigo fuente para de esta manera asegurar facilidad de mantenimiento.

2.3 Valida para la iteración o sprint

Se realizará esta actividad luego cada iteración.

2.4 Actividades

1. Identificar las secciones del código agregadas o cambiadas.
2. Ejecutar el análisis del código para detectar algún problema con el estándar establecido.
3. Establecer tarea en scrum para reparar inconsistencias si fueron encontradas.
4. Reparar las inconsistencias encontradas.
5. Si hubo inconsistencias, agregar los cambios realizados a la bitácora de cambios.

2.5 Encargado

Yordan Jiménez

2.6 Herramienta de ayuda

Checkstyle

3 Configuración del repositorio

3.1 Descripción de la actividad

El manejo de cambios en el código es crucial en un proyecto de desarrollo. Se debe asegurar que un cambio hecho pueda ser reversible para no poner en riesgo el sistema en ningun momento, y así poder tener respaldos en caso de que un cambio realizado no tenga consecuencias malas.

3.2 Propósito

Establecer pautas para el manejo de las versiones del proyecto en una herramienta de versionamiento, en este caso GitHub.

3.3 Valida para la iteración o sprint

A lo largo de todo el proyecto, antes de realizar un commit.

3.4 Actividades

1. Cada persona del equipo de desarrollo tendrá su propio branch del master, en el cual trabajara los cambios que realice en el software siguiendo el estandar aqui definido.

2. Cuando un desarrollador este seguro que su branch esta listo para ser implementado (esto es, ha pasado todas las actividades de calidad), enviara un pull-request para que su branch haga merge con el master.

3. En el Summary de los commits del branch de cada desarrollador debe haber un titulo que resuma el cambio implementado. En la seccion de Commentary es necesario comentar brevemente en lo que se trabajo y los cambios que se realizaron.

4. Los merge al master deberan tener el versionamiento adecuado según el estándar definido mas adelante.

5. El encargado de la actividad se compromete a periódicamente revisar los commits hechos por el equipo.

6. Cualquier Merge hecho al master debe quedar documentado en la Bitacora de Cambio, debe incluir los cambios realizados descritos en alto nivel.

3.5 Encargado

Fernando Molina

3.6 Herramienta de ayuda

GitHub.

4 Cambios en los artefactos de software

4.1 Descripción de la actividad

La consistencia de los artefactos de software es importante en el desarrollo de cualquier proyecto software. Es vital tener toda la informacion actualizada en todo momento, para que de esta manera los encargados de realizar tareas tengasn toda la informacion disponible y que no entre en conflicto lo que estipula un artefacto con la implementacion de otro.

4.2 Propósito

Asegurarse que los cambios en los artefactos de software sean documentados de manera periodica y oportuna.

4.3 Valida para la iteración o sprint

Cada vez que se realiza un cambio a un artefacto de software, y al final de cada sprint.

4.4 Actividades

1. Cuando hay un cambio en algún artefacto de software, debe ser documentado en la Bitácora de Cambio.
2. Cuando se realiza un commit al master, este debe incluir todos los artefactos de software de dicha versión, actualizados.
3. Si se encuentra alguna inconsistencia en los artefactos, esta debe ser reportada inmediatamente al encargado de esta actividad.
4. Cuando se realicen cambios a un documento, es necesario utilizar la funcionalidad de versiones de la herramienta Overleaf para facilitar la recuperación de versiones pasadas de los documentos.

4.5 Encargado

Leonardo Mendoza

4.6 Herramienta de ayuda

Overleaf, GitHub.

5 Cumplimiento del diseño respecto a los requerimientos

5.1 Descripción de la actividad

Para el usuario, el cumplimiento de los requerimientos funcionales es la parte mas importante del proyecto, el fin que se quiere cumplir. Es por ello que el diseño debe reflejar los requerimientos funcionales establecidos en el documento de requerimientos del sistema.

5.2 Propósito

Asegurarse que el diseño del sistema implementado cumple con los requerimientos funcionales planteados por el cliente.

5.3 Valida para la iteración o sprint

Al final de cada sprint.

5.4 Actividad

1. Por cada requerimiento funcional, revisar que en el diseño se encuentre un método o conjunto de métodos que cumplan la funcionalidad requerida.

5.5 Encargado

Leonardo Mendoza

5.6 Herramienta de ayuda

Ninguna.

6 Cumplimiento de las pruebas unitarias y de integracion

6.1 Descripción de la actividad

Las pruebas son una parte fundamental del aseguramiento de la calidad. La actividad consiste en realizar una serie de pruebas unitarios y de integración para asegurar el correcto funcionamiento del sistema a lo largo de todo el desarrollo, asi como garantizar que los cambios realizados en algún sprint no hayan afectado la funcionalidad del sistema.

6.2 Propósito

Se busca asegurar que el sistema se comporte de la manera esperada en todo momento por medio del desarrollo de pruebas unitarias y de integración que ayuden a comprobar el funcionamiento esperado del sistema.

6.3 Valida para la iteración o sprint

Se realizaran todas las pruebas antes de realizar un commit a la herramienta de control de versiones.

6.4 Actividades

1. Definir nuevas pruebas unitarias y de integracion a todos los modulos que fueron cambiados.

2. Realizar todas las pruebas de integración y unitarias.

3. Documentar en la bitácora de cambios las modificaciones realizadas a las pruebas.

4. Documentar en la bitácora de cambios el resultado de las pruebas realizadas.

6.5 Encargado

Yordan Jimenez.

6.6 Herramienta de ayuda

JUnit.