Avance 1 proyecto: Sistema de Análisis de Videos de Fútbol - Segmentación y clasificación no supervisada de jugadores.

Jose Garita Mario Naranjo Manuel Calderón

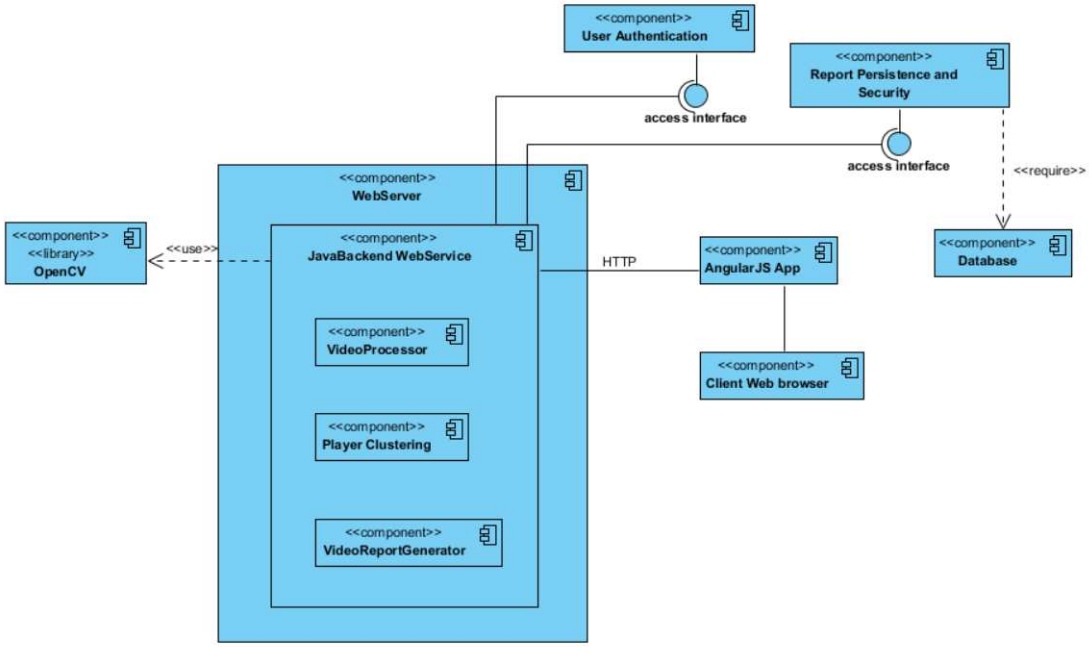
01-09-16

Contents

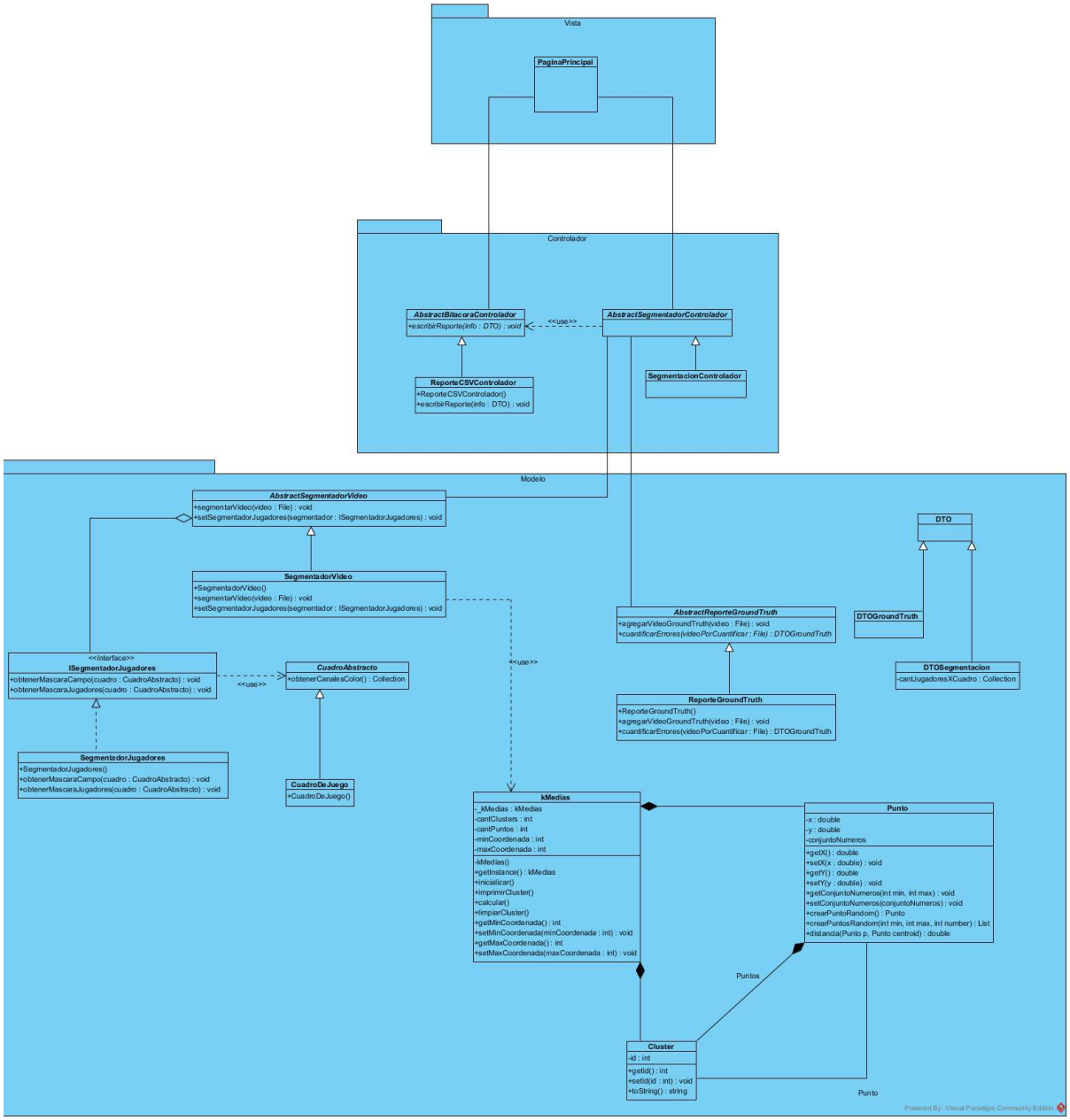
1 Estándares de codificación

2 Diagramas

2.1 Diagrama de componentes



2.2 Diagrama de clases



2.3 Patrón elegido:

Singleton

2.4 Razón:

Los patrones creacionales son patrones que se implementan al crear objetos en este caso lo único que se usa es en la creación de nuevos videos, pero estos no se almacenan, solo se procesan, por lo tanto la única creación es al iniciar el proceso de los videos, entonces un patrón creacional no llega al caso. No obstante el algoritmo kMedias es un algoritmo único que solo debe ser y analizarse en una instancia y no tener varias kmedias corriendo, por lo tanto se va a implementar por un patrón de instancia única llamado Singleton.

Los patrones comportamiento, se deben a la responsabilidades de los objetos, sus clases y los algoritmos que posee, pero en este caso volvemos a caer al mismo detalle:

- Chain of Responsibility: no existe la clases con responsabilidades similares y este lleve mensajes entre ellas, por lo tanto este patrón se descarta
- Interpreter: el recibir elemento del exterior y escribir, de igual manera no se ocupa más que retornar los videos y recibir uno de carga. No es necesaria un interprete.
- Iterator: la compisición de elementos por su algoritmo no es tal, cada clase varía en su objetivo y no compartes funcionalidades.
- Mediator : este es similar al interprete, la idea es que se comuniquen entre clases. Patrón no útil.
- Memento: no lleva propiedades de momentos que se se puede regresar, solo procesa y descarga los videos por cada fase.
- Observer: se puede implementar para cuando se termina de procesar el video, al acabar procesarlo, debe notificar a la interfaces web. No obstante la interfaz gráfica es sólo una y no hay por el momento lugar en este avance.
- Los demás patrones sucede igual, estos solo era para dar una pequeñas explicaición del porque no los elegimos.

Los patrones estructurales son de comspisición dentro de clases. Estos patrones fueron analizados en el grupo de trabajo y se concluye que ninguno es factible hasta este momento.

2.5 Bibliografias:

"Patrón De Diseño." - Wikipedia, La Enciclopedia Libre. N.p., 30 May 2016. Web. 28 Ago. 2016.

3 Actividades de evaluación de aseguramiento de la calidad

A continuación se presentan las actividades para el aseguramiento de la calidad del sistema de análisis automatizado de videos de fútbol: segmentación y clasificación no supervisada de jugadores. Estas actividades se realizan al finalizar cada <<sprint>>

3.1 Conformidad del diseño

- Realizar la trazabilidad de cada módulo, aparato o componente de diseño del sistema con cada requerimiento.
- Realizar y aplicar un <<checklist>> para evaluar el diseño realizado (evaluar la claridad del modelo) http://users.csc.calpoly.edu/~jdalbey/205/Deliver/designQAchecklist.ht
- Verificar que los diagramas y documentos de diseño y la matriz de trazabilidad están preparadas y mantenidas actualizadas, consistentes.

3.2 Conformidad de la implementación respecto al diseño

- Trazar cada artefacto del diseño con el código que lo implementa.
- Revisar la consistencia del código con el modelo (diseño)

3.3 Conformidad de la implementación respecto a los requerimientos

- Trazar cada requerimiento con el bloque de código fuente que lo implementa.
- Revisar la existencia de módulos implementados para la segmentación de videos.
  - Verificar la implementación correcta y/o parcial de cada algoritmo de segmentación (obtención de máscara de la cancha y <<blobs>> de jugadores)
- Revisar la existencia de módulos implementados de captura y procesamiento de videos (abrir, extracción de <<frames>>,y guardar video)
- Realizar revisión y cambios en la implementación. Registrar el cambio y las pruebas unitarias realizadas exitosas antes de subir al repositorio.
- Cumplimiento de estándares

3.4 Cumplimiento de estándares

- Realizar revisiones de código fuente y el código desarrollo para los cambios con el fin de detectar el uso incorrecto del estándar y la mantenibilidad del código.
- Generar una lista de comprobación para la inspección de código. http://users.csc.calpoly.edu/~jdalbey/20
- Identificar código poco auto-explicable y apuntar en caso contrario qué código es confuso.

3.5 Revisión de los tests unitarios, de integración y aceptación

- Diseñar y crear las pruebas unitarias, de integración. Crear un plan de seguimiento para aplicar las pruebas.
- Diseñar los casos de prueba de aceptación para cada requerimiento funcional y no-funcional.
- Reunirse, con acuerdo mutuo de tiempo, con el cliente para realizar pruebas de aceptación.
- Aplicar las pruebas de integración a los módulos o componentes del sistema (únicamente el código aprobado con las pruebas unitarias pueden ser ingresadas al repositorio)
- Generar un reporte resumen con resultados de pruebas del sistema.
- Documentar y reportar los errores/defectos encontrados en la aplicación de las pruebas.

3.6 Recolección y análisis de evidencias de defectos y errores

- Realizar inspecciones de código crítico (funciones, métodos y algoritmos importantes para procesamiento de videos) para detectar potenciales defectos.