



Aseguramiento de la Calidad Software IC-6831

Segundo Avance: Futbolito

Profesor: M. Sc. Saúl Calderón Ramírez

Estudiantes:

Carlos M. Girón Alas - 2014113159

Julián J. Méndez Oconitrillo - 2014121700

Daniel A. Troyo Garro - 2014073396



15 de octubre, 2016.

Índice

1	Documento de Requerimientos ISO/IEC/IEEE 29148-2011	1
1.1	Propósito del sistema	1
1.2	Alcance del sistema	1
1.3	Panorama general del sistema.	2
1.3.1	Contexto del sistema	2
1.3.2	Funciones del sistema	2
1.3.3	Características del usuario	3
1.4	Requerimientos funcionales	3
1.4.1	Acceso a la aplicación (Prioridad Media)	3
1.4.2	Subida del video (Prioridad Alta)	3
1.4.3	Envío del video a análisis (Prioridad Media)	4
1.4.4	Servidor web (Prioridad Media)	5
1.4.5	Algoritmos de segmentación temporal (Prioridad Alta)	5
1.4.6	Obtención de resultados (Prioridad Media)	5
1.4.7	Subida del archivo de ground truth (Prioridad Alta)	6
1.4.8	Análisis de ground truth (Prioridad Alta)	7
1.5	Requerimientos de usabilidad	7
1.5.1	Entendibilidad	7
1.5.2	Operabilidad	8
1.6	Requerimientos de eficiencia	8
1.6.1	En cuanto a la carga y subida de los elementos	8
1.6.2	En cuanto a la disponibilidad espacial	8
1.6.3	En cuanto a la disponibilidad de procesamiento	8
1.6.4	Tiempo de vida mínimo esperado	8
1.6.5	La duración esperada de una sesión	8
1.7	Interfaces del sistema	8
1.7.1	Navegación del sitio	8
1.7.2	Disposición espacial de los elementos en la interfaz	9
1.8	Operaciones del sistema	9
1.8.1	Requerimientos de integración del sistema humano	9
1.8.2	Mantenibilidad	9
1.8.3	Confiabilidad	10
1.9	Modos y estados del sistema	10
1.10	Características físicas	11
1.10.1	Requerimientos físicos	11
1.10.2	Requerimientos de adaptabilidad	11
1.11	Condiciones de ambiente	12
1.12	Seguridad del sistema	12
1.13	Administración de la información	12
1.14	Regulaciones y políticas	12
1.15	Sostenibilidad del ciclo de vida del sistema	13
1.16	Empaquetado, manejo, envío y transporte	13
1.17	Verificación	13
1.18	Dependencias y supuestos	14
2	Estándares de Codificación	15
2.1	Herramienta de verificación CheckStyle.	15
3	Validación del diseño	19
3.1	Jenkins: Tutorial de configuración	22
4	Versión actual del sistema ¿Cómo obtenerla?	34
5	Métricas	35
6	Pruebas Unitarias	39

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

1.1. Propósito del sistema

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

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

1.2. Alcance del sistema

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

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

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

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

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

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

1.3. Panorama general del sistema.

1.3.1. Contexto del sistema

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

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

1.3.2. Funciones del sistema

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

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

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

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

1.3.3. Características del usuario

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

1.4. Requerimientos funcionales

Los requerimientos funcionales se definen a continuación.

1.4.1. Acceso a la aplicación (Prioridad Media)

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

1.4.2. Subida del video (Prioridad Alta)

Compuesta de las siguientes etapas:

1.4.2.1 Opción de subir o reemplazar: (Prioridad Media)

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

1.4.2.2 Tiempo de espera: (Prioridad Baja)

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

1.4.2.3 Subida del video: (Prioridad Alta)

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

1.4.2.4 Visualización del video: (Prioridad Media)

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

1.4.2.5 Errores: (Prioridad Baja)

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

1.4.3. Envío del video a análisis (Prioridad Media)

Compuesta de las siguientes etapas:

1.4.3.1 Opción analizar: (Prioridad Media)

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

1.4.3.2 Selección de punto de corte manual: (Prioridad Baja)

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

1.4.3.3 Tiempo de espera: (Prioridad Baja)

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

1.4.3.4 Errores: (Prioridad Baja)

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

1.4.4. Servidor web (Prioridad Media)

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

1.4.5. Algoritmos de segmentación temporal (Prioridad Alta)

Los resultados que presenta el sistema una vez analizado el video provienen de un análisis de histogramas obtenidos de cada cuadro o *frame* en los que es dividido el video que se subió para ser analizado. Todos estos algoritmos son implementados por el programa de java ejecutándose en el servidor con la ayuda de funciones propias de la biblioteca *OpenCV*. De esta forma, se presentan a continuación dichas funciones.

1.4.5.1 Algoritmo de Bhattacharyya (Prioridad Alta)

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

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

1.4.5.2 Teorema de los tres sigmas (Prioridad Alta)

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

1.4.6. Obtención de resultados (Prioridad Media)

Una vez completado el proceso de análisis, el sistema cambiará su configuración para presentar los resultados obtenidos. Ahora, en lugar de presentarse los elementos iniciales, se presentarán un

botón de avance y un botón de retroceso que permiten al usuario avanzar entre los frames procesados, y se muestra una etiqueta que le indica al usuario si en ese frame se presentó un corte. Se considera un corte a aquel frame justo después de uno con una escena completamente diferente. Compuesta de las siguientes etapas:

1.4.6.1 Obtención de escenas ordenadas por posición: (Prioridad Media)

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

1.4.6.2 Visualización de cortes detectados (Prioridad Media)

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

1.4.6.3 Visualización de escenas (Prioridad Media)

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

1.4.6.4 Muestreo de información obtenida del análisis: (Prioridad Baja)

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

1.4.6.5 Descarga del informe en distintos formatos: (Prioridad Baja)

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

1.4.7. Subida del archivo de ground truth (Prioridad Alta)

Compuesta de las siguientes etapas:

1.4.7.1 Opción de subir o reemplazar: (Prioridad Alta)

La interfaz debe tener un botón claramente identificable que le permita subir un archivo de ground truth al servidor sin la necesidad de hacer ningún estilo de comprobación manual por parte del usuario.

1.4.7.2 Opcionalidad del Ground Truth (Prioridad Media)

La subida del archivo de ground truth es opcional, junto con el envío del video (que no es opcional). Al enviar un archivo a esa dirección, comienza un análisis con el servidor que devolverá si el análisis del video es satisfactorio respecto a lo especificado por los archivos de

1.4.7.3 Extensiones del Ground Truth (Prioridad Baja)

Los archivos válidos para el archivo de ground truth son únicamente .xml, .json, .csv, .yaml, .txt

1.4.8. Análisis de ground truth (Prioridad Alta)

Compuesta de las siguientes etapas:

1.4.8.1 Presentación del análisis comparativo: (Prioridad Alta)

Presentará la comparación entre los resultados obtenidos y los datos introducidos por el ground truth para generar una cuantificación de falsos positivos y negativos.

1.4.8.2 Formato del archivo: (Prioridad Baja)

La forma en la que esté diseñado manualmente el archivo de ground truth debe seguir el formato json para mostrar el frame en el que ocurrió el cambio, el número de frame y la cantidad de frames que hubo entre cortes. El formato mínimo que debe aceptarse es .csv parseado de la siguiente forma:

frame inicial, fram final, tipoCorte, tipoEscena

Las columnas frame inicial y frame final deben ser valores enteros positivos, puesto que indicarán el número de frame en el que inicia y termina una escena. Las columnas adicionales no se tomarán en cuenta para el muestreo de resultados.

1.5. Requerimientos de usabilidad

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

1.5.1. Entendibilidad

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

1.5.1.1 Mensajes de éxito, advertencias y errores

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

1.5.1.1.1 Error

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

1.5.1.1.2 Advertencia

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

1.5.1.1.3 Éxito

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

1.5.2. Operabilidad

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

1.6. Requerimientos de eficiencia

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

1.6.1. En cuanto a la carga y subida de los elementos

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

1.6.2. En cuanto a la disponibilidad espacial

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

1.6.3. En cuanto a la disponibilidad de procesamiento

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

1.6.4. Tiempo de vida mínimo esperado

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

1.6.5. La duración esperada de una sesión

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

1.7. Interfaces del sistema

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

1.7.1. Navegación del sitio

Se compone de los siguientes elementos

1.7.1.1 Flujo natural de la página

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

1.7.1.2 Retroceso

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

1.7.2. Disposición espacial de los elementos en la interfaz

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

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

1.8. Operaciones del sistema

Se subdividen en los siguientes elementos.

1.8.1. Requerimientos de integración del sistema humano

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

1.8.2. Mantenibilidad

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

1.8.2.1 Tiempo

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

1.8.2.2 Ritmo

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

1.8.2.3 Complejidad del Mantenimiento.

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

1.8.2.4 Índices de acción de mantenimiento

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

1.8.3. Confiabilidad

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

1.9. Modos y estados del sistema

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

- **1.9.1 Estado sin sesión:** En este estado la aplicación web se mantiene en la pantalla de inicio de sesión, aguardando que el usuario ingrese los datos para iniciar su sesión y que puede operar la plataforma. En este modo no se permiten acciones adicionales al ingreso de los datos de seguridad para el inicio de sesión.
- **1.9.2 Manejo de cuenta y de la plataforma:** En este estado se le permite al usuario ver los informes generados de anteriores videos cargados, así como demás información que sea considerada importante como la fecha de subida y el usuario que subió el video. Además se le permite al usuario el cambio de datos de su cuenta tales como el nombre de usuario o la contraseña.
- **1.9.3 Modo de espera de carga del video:** En este estado el sistema se encuentra a la espera de que el usuario cargue a la página un video para así poder posteriormente procesarlo y segmentarlo. No se muestra ninguna información de resultados relevante y es el modo más importante del sistema al ser el necesario para el ingreso de información y datos. El sistema mostrará mensajes de error si se trata de subir un video no válido (formato no aceptado o errores similares) o un archivo corrupto.
- **1.9.4 Modo de selección de cortes y visualización de video subido:** El estado en el que entra el sistema tras de que se detecte y cargue el video inicial. En este modo el usuario puede visualizar el video subido, subir un archivo *ground truth* con la información de los cortes o escenas deseadas y además se le permite al usuario seleccionar los puntos entre frames del video donde se realizará un corte de escena. El usuario podrá pasar aquí al modo de Estado de procesamiento al enviar el video.
- **1.9.5 Estado de procesamiento:** Estado en el que se muestra el porcentaje carga del sistema respecto a la carga del video o al análisis total del video, así como la generación de informes. En este estado únicamente se le muestra al usuario el valor de completitud de la tarea que actualmente esté realizando el sistema.

- **1.9.6 Muestreo de resultados:** Tras realizar un análisis exitoso de un video subido al sistema, este devolverá una serie de resultados tales como el número de escenas obtenidas, el momento en el video en el que termina y comienza cada escena, la visualización de cada escena y demás datos relacionados al rendimiento del equipo y de cada jugador a nivel individual. Se le permitirá al usuario archivar estos resultados para que sean consultados después por él o demás miembros de la plataforma. También se le permitirá descargar informes generales de los resultados obtenidos del video en distintos formatos. Es el estado que más información brinda de los datos pertenecientes al sistema, junto con *Manejo de cuenta y de la plataforma*.

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

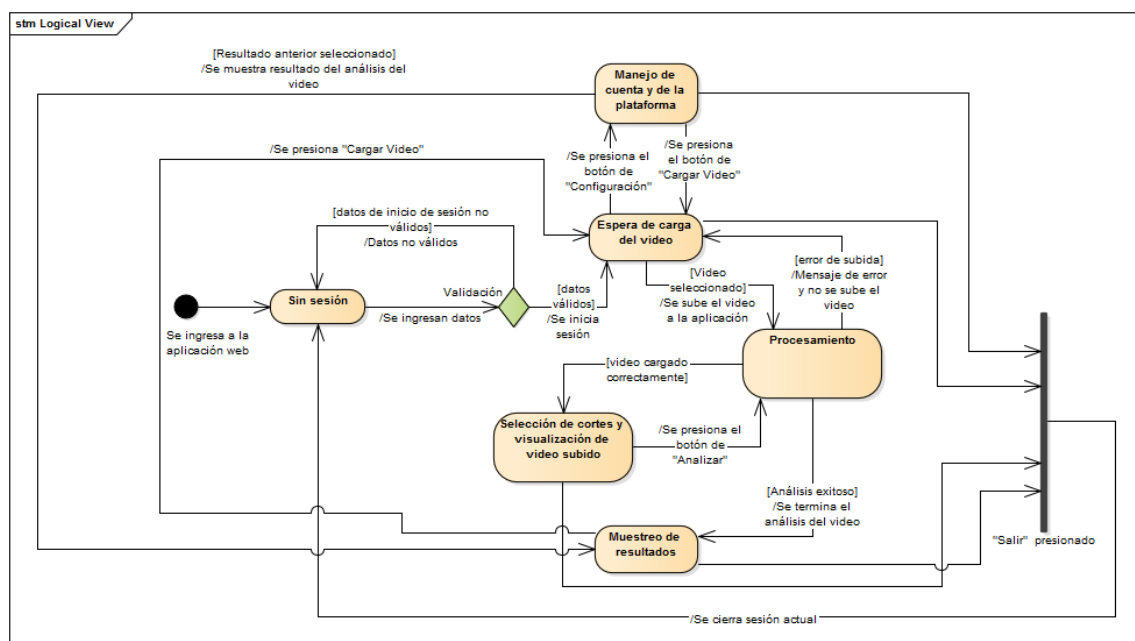


Figura 1. Diagrama de estados

1.10. Características físicas

Las características físicas del producto se describen a continuación.

1.10.1. Requerimientos físicos

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

1.10.2. Requerimientos de adaptabilidad

El sistema se estima que en un futuro podría integrarse como un módulo de un sistema de trabajo de videos para encontrar patrones o analisis extensivo de información en grandes cantidades.

Así, se considera que deben ser completamente intercambiables las funciones que analizan los archivos de video así como aquellas que generan las estadísticas y datos de valor respecto al contenido analizado.

1.11. Condiciones de ambiente

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

1.12. Seguridad del sistema

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

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

1.13. Administración de la información

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

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

1.14. Regulaciones y políticas

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

Se requerirá que cada cuenta esté asociada a un miembro del cuerpo técnico del club como de la junta administrativa, por lo que dichos datos serán requeridos a la hora de ingresar una cuenta y

validados por el equipo de soporte del sistema. Estos datos serán albergados en la base de datos de manera encriptada, por lo que permanecerán seguros a posibles filtraciones o robos de información. Adicional a estas regulaciones ninguna otra política es requerida por parte del equipo de desarrollo como del club Boca Juniors.

1.15. Sostenibilidad del ciclo de vida del sistema

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

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

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

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

1.16. Empaquetado, manejo, envío y transporte

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

1.17. Verificación

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

Las pruebas de aceptación serán realizadas emulando los casos de uso en el que podrá estar el usuario al utilizar el sistema en su totalidad. Así, un miembro del equipo de desarrollo emulará dicho caso y ejecutará todas las funcionalidades de la misma manera de la que lo haría un usuario

promedio. Si todo el proceso es exitoso, se considera que la prueba fue validada.

1.18. Dependencias y supuestos

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

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

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

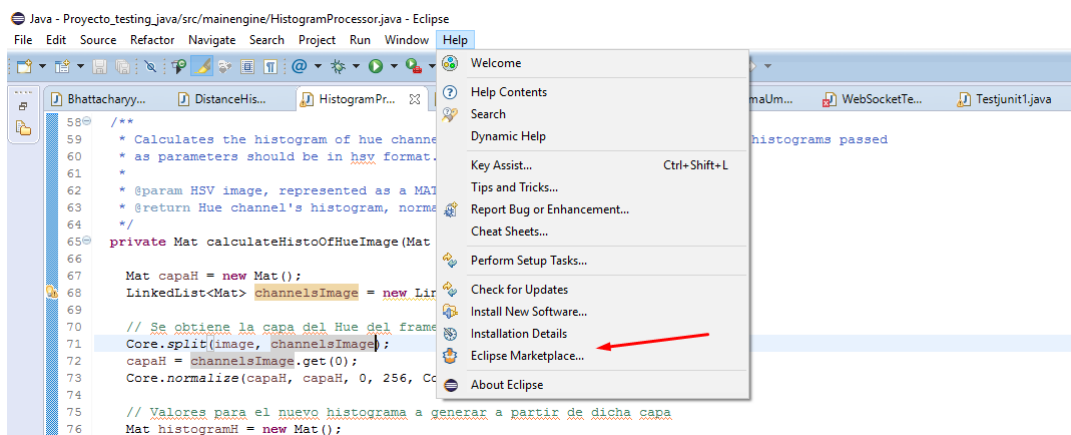
2. Estándares de Codificación

En avances anteriores, se habían escogido dos estándares de codificación para el presente proyecto. Estos estándares eran el estándar de codificación planteado por Google para el lenguaje java llamado *Google Java Style* y el estándar de codificación para javascript brindado por la herramienta en línea **jshint.com**. Estos dos estándares han sido implementados en el código fuente del proyecto desde entonces, siendo el estándar de Java implementado automáticamente mediante la herramienta o *plugin* de Eclipse **CheckStyle**. A continuación, se muestra un tutorial sobre cómo instalar y utilizar esta herramienta para automatizar la conformidad del estándar en el proyecto.

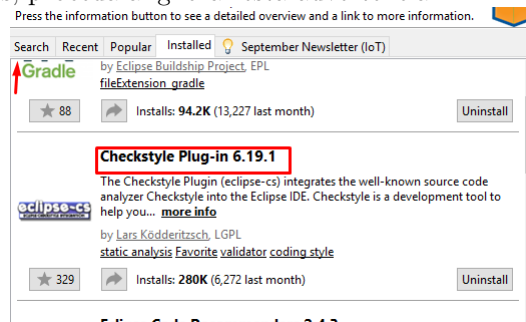
2.1. Herramienta de verificación CheckStyle.

Checkstyle es una herramienta automatizada de revisión de código estático, especial para entornos en donde se utiliza el lenguaje Java. Permite a los programadores revisar si su código java se adhiere a un estándar específico, entre los cuales tiene como opciones predeterminadas al estándar de codificación de *Sun Code Conventions* y el *Google Java Style*, el cual es el elegido en este caso. El método para instalación de esta herramienta y su utilización en el entorno de desarrollo Eclipse se explica a continuación:

- Al tener el entorno de desarrollo abierto, diríjase a la pestaña **Help** o **Ayuda**, en la parte superior de la ventana y seleccione la opción **Eclipse Marketplace**. La imagen muestra la localización de esta opción.

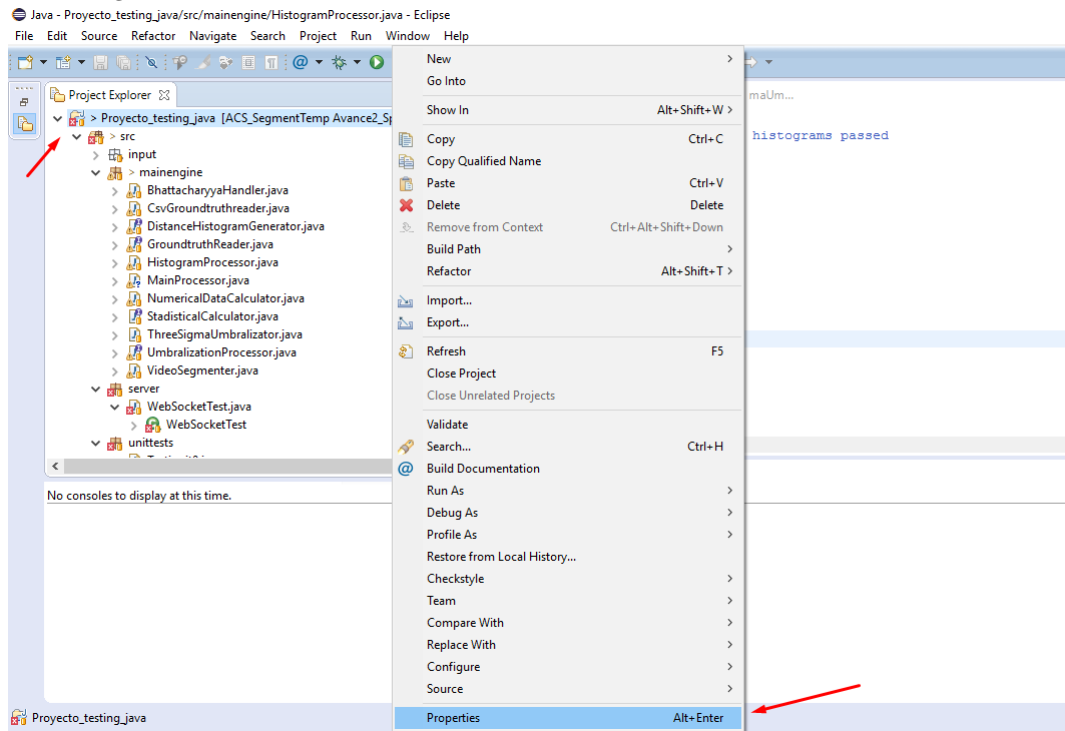


- Al abrirse la ventana, diríjase a la ventana **Search** e introduzca *Checkstyle* en el espacio de búsqueda. Podrá ver que el buscador lanzará como resultado el **Checkstyle Plug-in**. Seleccione el botón de Instalar y siga el proceso de instalación, el cual le pedirá reiniciar Eclipse al finalizar. En la imagen se muestra la pestaña de **Installed** y el botón de **Uninstall** en vez del de **Install** al estar el plugin ya instalado en la terminal. Si lanza una ventana de advertencia durante el proceso de instalación diciendo que se está instalando software de fuentes desconocidas, proceda a ignorar esta advertencia.

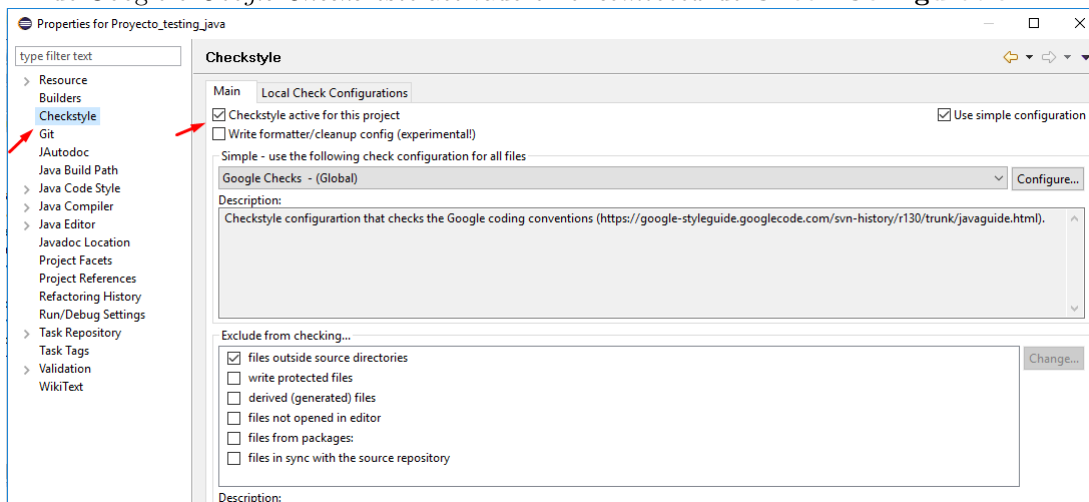


- Tras reiniciar Eclipse, deberíamos contar con la herramienta de Eclipse ya instalada en nuestro entorno local de desarrollo. Ahora basta activarlo en el proyecto para poder utilizarla. Para

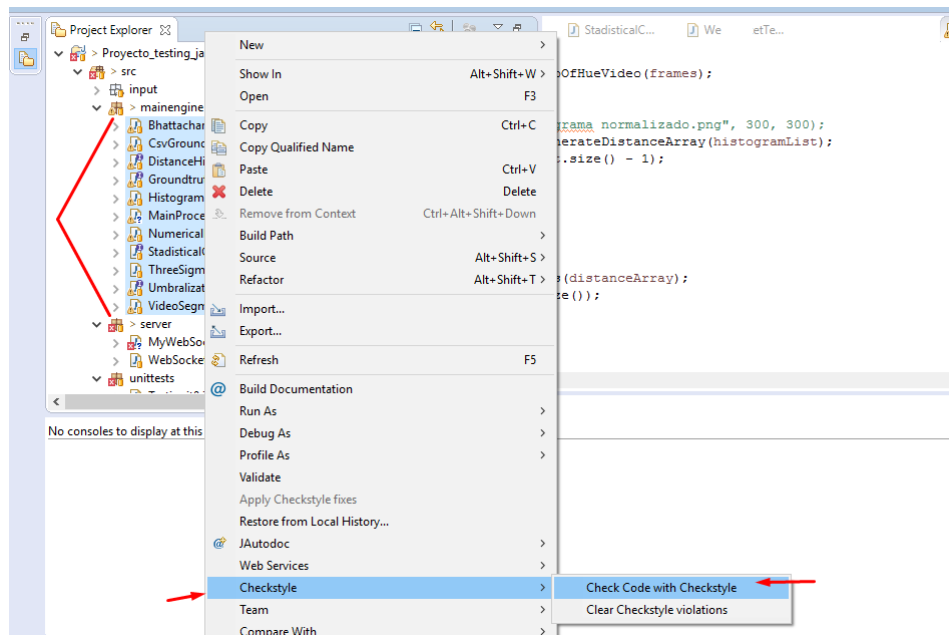
hacer esto diríjase al *Project Explorer* y realice click derecho sobre el proyecto, el cual debería estar en el *Workspace* para este momento. Diríjase al botón **Properties**, tal y como indica la imagen.



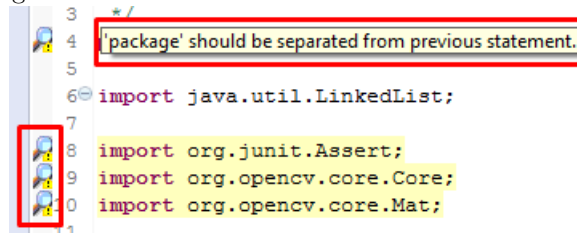
- En la barra lateral, asegúrese de entrar al menú de **CheckStyle** y una vez dentro, active la opción **Checkstyle active for this project**. Esto debería permitir utilizar la herramienta en el proyecto. La imagen indica la posición de estas opciones. Asegúrese de que el estándar de Google o *Google Checks* esté activado en el *combobox* de **Check Configuration**.



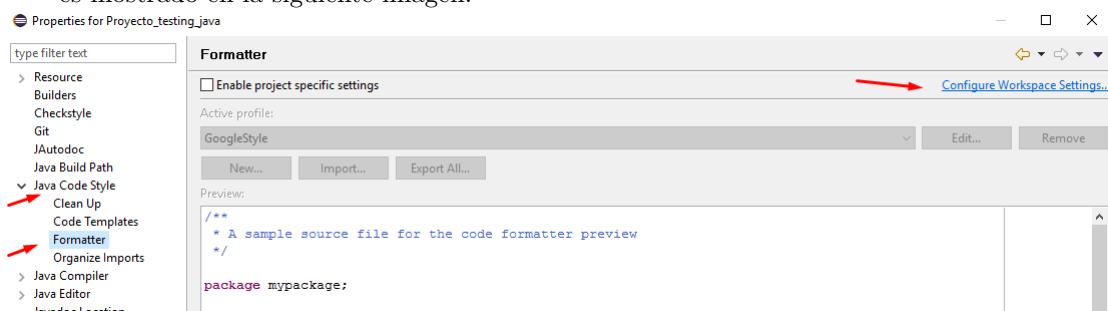
- Para poder utilizar la herramienta de CheckStyle, seleccione click derecho en el *Project Explorer* los archivos de código fuente que desea verificar y en la opción de **CheckStyle**, haga click sobre **Check code with CheckStyle**. Esto debería activar un chequeo de implementación del estándar seleccionado.



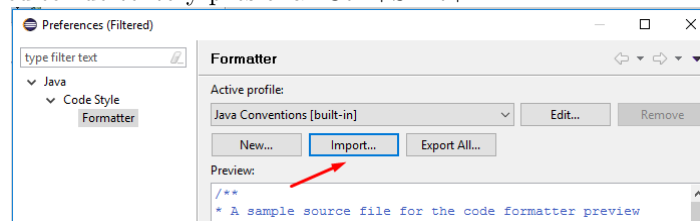
- Dicho chequeo mostraría con una lupa al lado izquierdo del editor de texto del entorno de desarrollo Eclipse aquellas líneas del archivo de código fuente que cuenten con violaciones al estándar de codificación *Google Java Style*. Al mantener el cursor sobre dicho símbolo, un *tooltip* mostrando el tipo de violación específica aparecerá. Todo esto se muestra en la siguiente imagen.



- CheckStyle es una herramienta únicamente de verificación, lo que indica que únicamente revisará el código, mas no lo corregirá. Eclipse cuenta con un formateador de código, siguiendo una configuración específica. Como punto extra a este tutorial, se enseñará a corregir de manera automática y rápida el código para que siga el estándar de codificación. Diríjase otra vez a las propiedades del proyecto haciendo click derecho sobre la carpeta del proyecto en el **Project Explorer** y seleccionando la opción **Properties**.
- Al abrirse la ventana de propiedades del proyecto, diríjase al menú **Java Code Style** y de ahí al submenú **Formatter**. En la carpeta raíz del repositorio debería haber un archivo llamado `eclipse-java-google-style.xml`. Asegúrese de que exista y que se encuentre en la copia local de su repositorio.
- Seleccione el botón de configuración **Configure Workspace Settings**. La posición del botón es mostrado en la siguiente imagen.



- Al abrirse la ventana, seleccione el botón de **Import** y seleccione el archivo .xml anteriormente referenciado. La imagen muestra la posición del botón. Una vez seleccionado dicho archivo y configurado el tipo de formateo de código al de *Google Java Style*, puede seleccionar el código en el editor de texto y presionar Ctrl+Shift+F.



3. Validación del diseño

En la siguiente tabla se muestran las relaciones entre las unidades de diseño y los requerimientos del sistema:

Artefacto de diseño	Requerimientos asociados
Diagrama de Casos de Uso	<p>Los requerimientos funcionales relacionados son.</p> <ul style="list-style-type: none">■ 1.4.1 de acceso a la aplicación.■ 1.4.2 para la subida del video (las opciones que tiene el usuario y lo que debe presentársele).■ 1.4.3 de envío del video para análisis pues muestra las opciones que tiene el usuario para iniciar el proceso de análisis).■ 1.4.6 por los subrequerimientos 1.4.6.2, 1.4.6.3, 1.4.6.4, 1.4.6.5.■ El requerimiento 1.4.7 por las opciones que tiene el usuario para subir el ground truth en la interfaz.■ 1.4.8 por la presentación de la información del ground truth (1.4.8.1) <p>Los requerimientos de usabilidad 1.5 están relacionados seriamente con los casos de uso por definición.</p>
Diagrama de Clases	<p>Inverso a los requerimientos relacionados al diagrama de casos de uso, estos se enfocan más en el código del servidor y los procesos internos del programa, por lo que se relaciona a los requerimientos funcionales:</p> <ul style="list-style-type: none">■ 1.4.3.3 y 1.4.3.4 por el manejo de errores y del tiempo de ejecución.■ 1.4.4 por el servidor pues se encarga de muestrear la forma en la que se relaciona con el resto de clases.■ 1.4.5 y todos sus subrequerimientos: 1.4.5.1 por la implementación del Bhattacharyya y 1.4.5.2 por el teorema de los tres sigmas pues muestra una forma inicial de cómo es que se implementa.■ 1.4.6 porque se muestra la forma inicial en la que se obtienen los resultados. <p>Pueden mapearse también a los requerimientos de eficiencia 1.6 en caso de que se quiera hacer un análisis particular de algún elemento o relación.</p>

Diagrama de Componentes	<p>Los requerimientos funcionales asociados son:</p> <ul style="list-style-type: none"> ■ 1.4.2 porque se muestra el proceso por el que pasan los videos que son subidos. Específicamente al requerimiento 1.4.2.3 ■ 1.4.4 porque se muestran de forma específica cuáles son los componentes relacionados al servidor y de qué forma se relacionan con los componentes del front end. ■ 1.4.6 pues da una idea general de la forma en que se relacionan los componentes para devolver los resultados y que se puedan presentar en el front end. ■ 1.4.7 pues se muestra o se da una idea, al igual que en la 1.4.2, cómo viaja el ground truth entre componentes. <p>Se relaciona hasta cierto punto a los requerimientos de eficiencia pues los componentes pueden ser analizados individualmente en búsqueda de mejoras de eficiencia particulares.</p>
Diagrama de Actividad	<p>Los requerimientos funcionales asociados son:</p> <ul style="list-style-type: none"> ■ 1.4.2, específicamente a 1.4.2.1 y 1.4.2.4 pues son la interacción que tendrá el usuario con el sistema. También relacionado está 1.4.2.5 por el muestreo de errores. ■ 1.4.3, especialmente por la 1.4.3.1 porque se muestra la opción con la que ha de interactuar el usuario para enviar el video, y los errores que puede recibir (1.4.3.4) ■ 1.4.6, enfocado en 1.4.6.2, 1.4.6.3, 1.4.6.4 porque es la interacción resultado para que el usuario vea lo que resolvió el sistema. ■ Es una tarea por sí misma relacionada también a 1.4.6, el requerimiento 1.4.6.5 muestra como el sistema debe permitir la interacción para obtener el archivo con los resultados. ■ 1.4.7, con 1.4.7.1 pues es la interacción mediante la cual el usuario sube un video al sistema. ■ 1.4.8, en especial de 1.4.8.1 para presentar el análisis comparativo. <p>Los requerimientos de usabilidad 1.5 están relacionados a este diagrama pues muestra la forma en que la actividad del usuario respecto al sistema.</p>

Diagrama de Estados	<p>Los requerimientos funcionales asociados son:</p> <ul style="list-style-type: none"> ■ Está relacionado de forma general a los requerimientos funcionales 1.4, porque el diagrama de estados permite visualizar la forma en la que los requerimientos se relacionan entre sí. ■ Además se relaciona generalmente a varios requerimientos de usabilidad 1.5 pues muestra la forma en que el sistema reaccionará de acuerdo al uso que se le dé.
---------------------	---

3.1. Jenkins: Tutorial de configuración

Jenkins es una herramienta de integración continua, que proviene de Hudson (es un fork de dicho proyecto) y que propone un sistema de fácil instalación y configuración. La portabilidad de esta herramienta representa un problema pues está pensada para Linux principalmente. La configuración en primer instancia debería ser sencilla en cualquier sistema operativo, pero por una situación con la generación de credenciales para el usuario de Jenkins, en Windows no fue posible configurarlo sin abrir una máquina virtual con un servidor de ubuntu y una referencia desde el servicio de Jenkins en Windows. Como esto es inaceptable y un verdadero desastre, para este tutorial, se muestra una versión únicamente para Ubuntu 16.04, en donde la dimensión del problema sigue siendo tolerable.

1. Primero, descargar Jenkins del sitio oficial: <http://pkg.jenkins-ci.org/debian-stable/>, en esta misma página encontrará instrucciones para descargar desde la terminal. Estas son:

- Ejecutar el primer comando: **wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key**
— **sudo apt-key add -**
- Ejecutar **sudo nano /etc/apt/sources.list** Nano es un editor de texto en consola que abrirá el archivo sources.list en esa ubicación.
- A ese archivo, agregar el comando **deb https://pkg.jenkins.io/debian-stable binary/** al final del archivo.
- Ejecutar **sudo apt-get update** y **sudo apt-get install jenkins** para comenzar la instalación.

De forma alterna (y recomendada), descargar la versión estable más reciente para instalación directa del .deb.

If you need *.deb for a specific version, use these.

Name	Last modified	Size
jenkins_2.7.4_all.deb	2016/09/08	64.8M
jenkins_2.7.3_all.deb	2016/08/31	65.3M

Y luego, desde la carpeta donde se descargó el .deb (click derecho Open Terminal), ejecutar

- **sudo dpkg -i nombrepaquete** (en este caso, **jenkins_version.all.deb**)
- La interpretación del comando anterior es para nuestro ejemplo: **sudo dpkg -i jenkins_2.7.4_all.deb**

En caso de que diera error por falta de un daemon, **sudo apt-get -f install** resolvería los problemas de la dependencia faltante.

2. En este punto, Jenkins fue descargado. Lo que sigue es instalarlo junto con los plugins requeridos para la inicialización de un proyecto con el repositorio.

The image shows the 'Getting Started' screen of Jenkins. The title is 'Unlock Jenkins'. Below it, a message states: 'To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server: /var/lib/jenkins/secrets/initialAdminPassword'. It then asks the user to 'Please copy the password from either location and paste it below.' There is a text input field labeled 'Administrator password'. At the bottom right, there is a blue 'Continue' button. A faint Android robot logo is visible in the background.

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

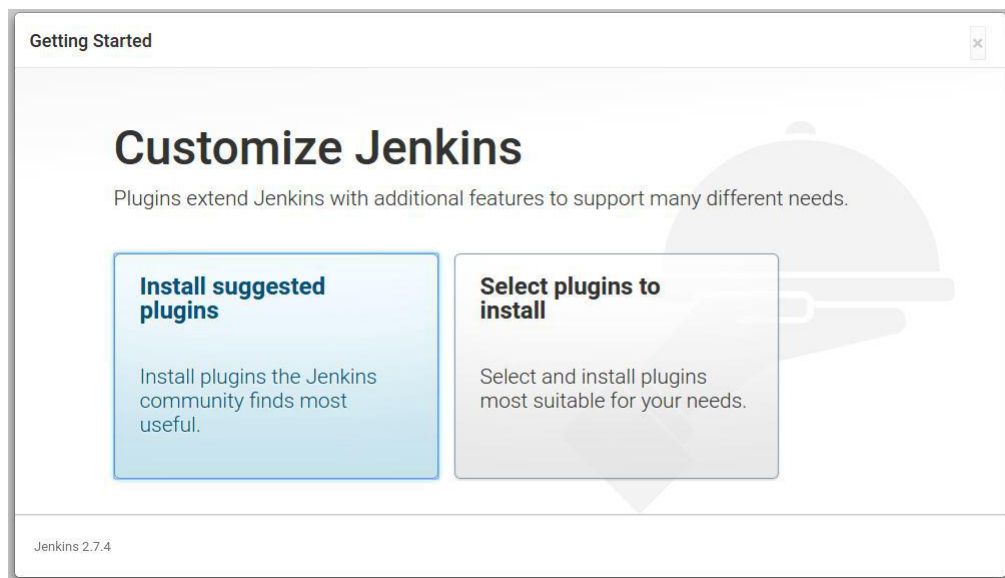
Administrator password

Continue

Si abre un browser (en este caso Google Chrome) y entra a localhost:8080, por defecto, el servicio de Jenkins corre en dicho puerto de localhost, le preguntará por una contraseña inicial.

Puede ejecutar **sudo cat /var/lib/jenkins/initialAdminPassword** para obtener directamente el contenido del archivo texto con la llave, o puede abrir manualmente el archivo y recuperar el texto. Después de copiarlo y pegarlo, dé click en Continue.

Una vez con acceso, le aparecerá esta opción. Escoja Install Suggested Plugins pues no hace falta complicarse de más.

The image shows the 'Getting Started' screen of Jenkins, specifically the 'Customize Jenkins' section. The title is 'Customize Jenkins'. Below it, a message states: 'Plugins extend Jenkins with additional features to support many different needs.' There are two main options presented in boxes: 'Install suggested plugins' (highlighted with a blue border) and 'Select plugins to install' (greyed out). The 'Install suggested plugins' box contains the text: 'Install plugins the Jenkins community finds most useful.' The 'Select plugins to install' box contains the text: 'Select and install plugins most suitable for your needs.' At the bottom left, it says 'Jenkins 2.7.4'. A faint Android robot logo is visible in the background.

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

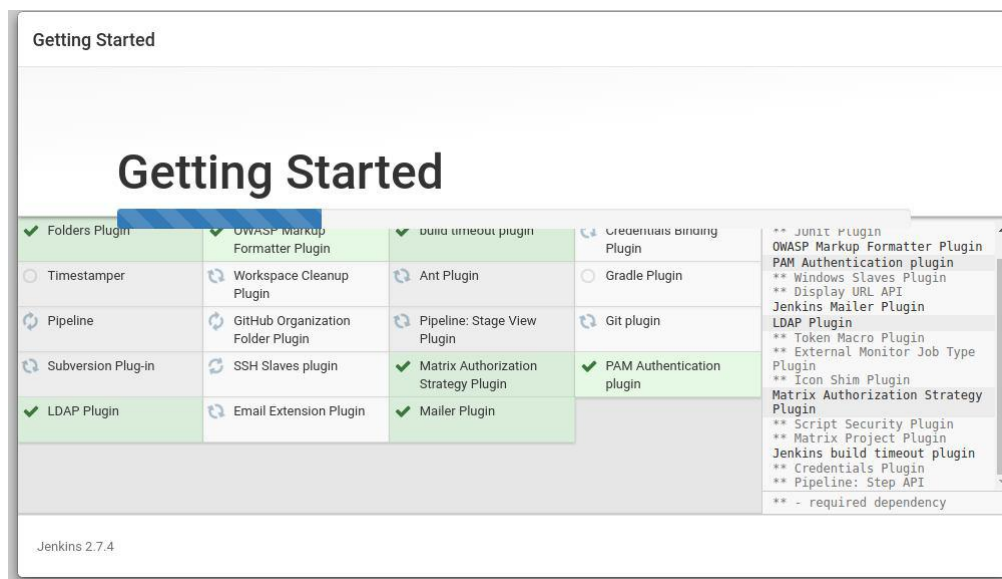
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.7.4

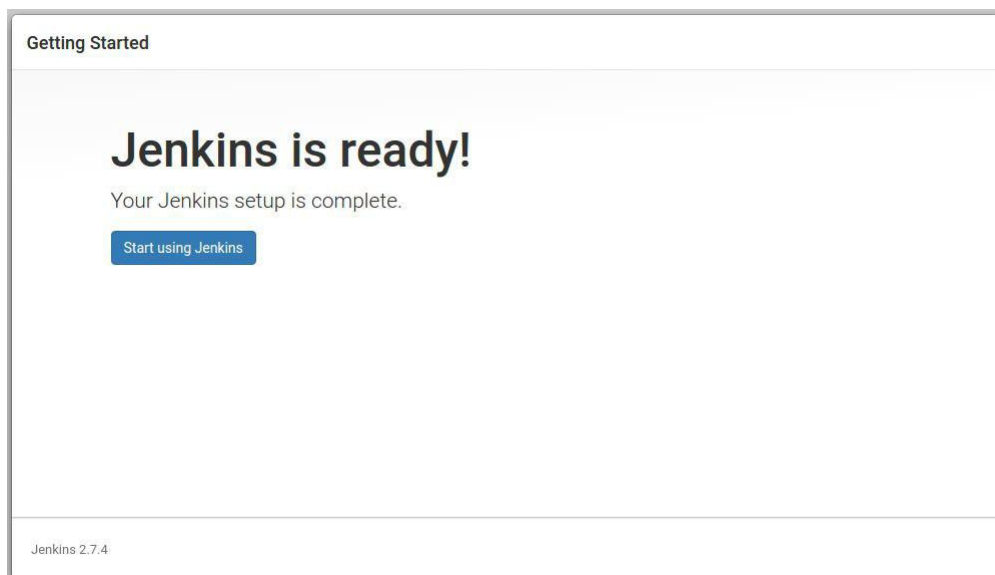
Le aparecerá una pantalla como la siguiente. Espere a que se complete el proceso de instalación.




La mayoría de tutoriales de fácil acceso en internet le indicarán la forma de instalar manualmente los plugins de Github y Git, sin embargo en la pantalla anterior se puede ver cómo ambos son plugins sugeridos para Jenkins (al menos, en la versión de este tutorial), por lo que no hará falta instalarlos de nuevo.

3. Ahora, configuraremos el usuario de uso de Jenkins. Este paso es vital pues Jenkins el resto del tutorial hará uso de estos datos, así que es recomendado tenerlos a mano.

Para este tutorial, diremos que el usuario es **menoc** y el password es cualquier cadena. Llene los demás datos como es evidente que debe hacerse, y dé click en Save and Finish para continuar. Aparecerá esta pantalla diciendo que el proceso fue exitoso.





Ahora, recargue o abra de nuevo Jenkins y obtendrá la siguiente pantalla principal en la que deberá iniciar sesión con el usuario que acaba de crear.


**Jenkins**


Julían Méndez | [log out](#)


Jenkins >


 New Item

 People

 Build History

 Manage Jenkins

 My Views

 Credentials

User:

Password:

☐ Remember me on this computer

[log in](#)

Build Queue


No builds in the queue.

Build Executor Status

1 Idle


2 Idle


Dé click en login y estamos listos.


**Jenkins**


Julían Méndez | [log out](#)


Jenkins >


 New Item

 People

 Build History

 Manage Jenkins

 My Views

 Credentials

Welcome to Jenkins!

Please [create new jobs](#) to get started.

Build Queue

No builds in the queue.

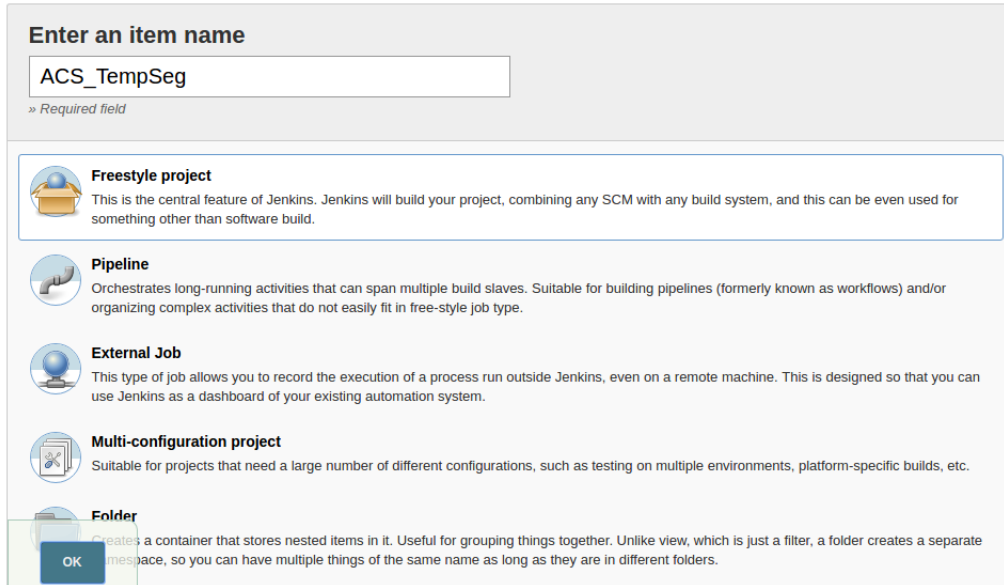
Build Executor Status

1 Idle

2 Idle

4. Configurar el Item de Jenkins con el proyecto de Github

En el último screenshot se mostraba al lado izquierdo la opción New Item. Dé click ahí para crear un nuevo Job. Como ya se dijo previamente, en la mayoría de tutoriales aparecen los pasos para instalar el plugin de Github y Git pero no hará falta. En caso de que estos plugins no estén instalados, debe entrarse a Manage Jenkins, Manage Plugins, buscar Github e instalarlo (traerá todos los plugins relacionados que requiere para funcionar).



Enter an item name

ACS_TempSeg

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Seleccione el nombre de su Job. En este caso, será ACS_TempSeg por el motivo del proyecto. Después, seleccione Freestyle project y luego OK para continuar.

La siguiente parte es el motivo por el que se trabaja en Linux (Ubuntu 16.04 en este caso). Siga con cuidado los siguientes pasos pues algún fallo complicará de forma ridícula la instalación.

- Abra la terminal y ejecute **sudo -i** para usar el usuario root
- Ejecute **apt-get install git -y**
- Ejecute **apt-get install apache2 -y**
- Ejecute **chown jenkins /var/www/html -Rf**

- Ejecute **visudo** lo que entrará a un editor de texto.

```

root@carlos-Aspire-E1-522: ~
GNU nano 2.5.3 File: /etc/sudoers.tmp

# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:$
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
[ Read 30 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line

```

- Ubíquese en la línea debajo de Defaults secure_path etc y escriba
Defaults:jenkins !requiretty,!lecture
jenkins ALL=NOPASSWD:/etc/init.d/apache2

```

root@carlos-Aspire-E1-522: ~
GNU nano 2.5.3 File: /etc/sudoers.tmp Modified

# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:$
Defaults:jenkins !requiretty,!lecture
jenkins ALL=NOPASSWD:/etc/init.d/apache2
# Host alias specification
# User alias specification
# Cmnd alias specification

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line

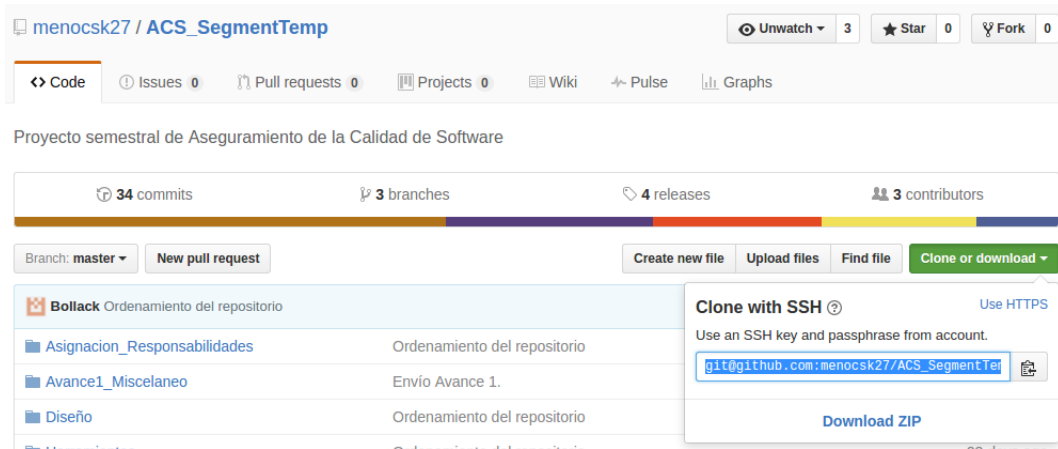
```

Salve el archivo con Ctrl + X, luego Y y por último Enter para confirmar.

- Esto asegura que apache está instalado, da permiso a Jenkins de copiar archivos al directorio web y permite reiniciar fácilmente apache.
- NO CIERRE ESTA TERMINAL AÚN.

Ahora que todos esos pasos fueron correctamente ejecutados, es importante hacer la aclaración de que en un proyecto a gran escala, apache corre en un server aparte que es accesado. Todos los pasos anteriores son simplemente para poder conectar correctamente el repositorio de forma local.

Ahora, veamos cómo configurar el repositorio. Entre a github y copie el ssh del repositorio que desea configurar.



En la terminal que estaba en root, haga lo siguiente. De nuevo, los siguientes pasos involucran comandos de terminal, por lo que debe tenerse cuidado.

- Ejecute **su jenkins** para cambiar al usuario de jenkins.
- Ejecute **cd** para cambiar al directorio home de jenkins.
- Ejecute **git clone git@github.com:menocsk27/ACS_SegmentTemp.git** o el ssh que del repo que se quiere clonar.

```
root@carlos-Aspire-E1-522:~# chown jenkins /var/www/html -Rf
root@carlos-Aspire-E1-522:~# visudo
root@carlos-Aspire-E1-522:~# su jenkins
jenkins@carlos-Aspire-E1-522:/root$ cd ~
jenkins@carlos-Aspire-E1-522:~$ git clone git@github.com:menocsk27/ACS_SegmentTemp.git
```

- Aparecerá un mensaje de error y luego una solicitud de confirmación, escribimos **yes** y ahora github es uno de los hosts conocidos.

```
jenkins@carlos-Aspire-E1-522:~$ git clone git@github.com:menocsk27/ACS_SegmentTemp.git
Cloning into 'ACS_SegmentTemp'...
The authenticity of host 'github.com (192.30.253.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGL7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.113' (RSA) to the list of known hosts.
Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
jenkins@carlos-Aspire-E1-522:~$
```

En este punto nos apareció un mensaje de error diciendo que la conexión no puede activarse, pero eso está bien. Lo que haremos ahora es generar un llave SSH de acceso a github para obtener esos permisos. Lo último que haremos en la consola antes de volver a la interfaz de usuario de Jenkins es generar esa llave.

- Ejecutamos `ssh-keygen -t rsa -b 4096 -C "menoc.sk27@gmail.com"` Donde el correo electrónico es el correo relacionado a la cuenta de Github. Pedirá un par de cosas más pero con dar Enter basta.

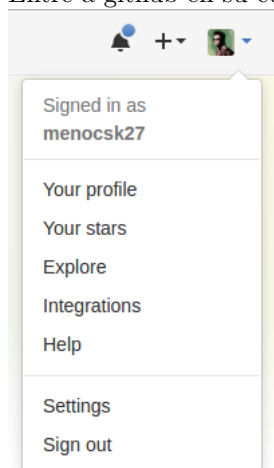
```
jenkins@carlos-Aspire-E1-522:~$ ssh-keygen -t rsa -b 4096 -C "menoc.sk27@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/jenkins/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/id_rsa.
Your public key has been saved in /var/lib/jenkins/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:PYAX1+bBgD02NSK/UjAly3xpuF1kbPTSILF/u60Ywuo menoc.sk27@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|  ..o+B+ |
| o+=*B=+= |
| **BBo++.. |
| *+* o.. . |
| ..S.+ . |
| . . . . |
| o . o . |
| o o . . |
| .E. .o . |
+---[SHA256]-----+
jenkins@carlos-Aspire-E1-522:~$
```

Con esto se generó la llave que usaremos para conectarnos al repositorio desde Jenkins

- Ejecute `cat ~/.ssh/id_rsa.pub` y se mostrará la llave generada.

```
jenkins@carlos-Aspire-E1-522:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC8Nzn6Mky/dsJ1kJPi1gksjhygfasqtqwv3IoogdZ
1TPWeNwgX6NGN+Ohd3681WYHrf+8Py2PUPCiJHx/Iq+TmtIp3Ptkhu+11LK9Qi1e/mudXmtCaGV9C9lE
uSzJzQLjs44fGQLnx9pIMazak+kevmD9iDX9/69z+pLkPM2U9ChTq2/gqBMQTcGRzHsJx1w3BAuySou0
aaNW9jjNhhs3nbpdk3oEb4smhXfd5Q0IC6C1f0e20K5Hfga4llzvm1sT2+St6VLxTCLwmdssb0yywR7L
scZzfQ5HeYJogOGbYWrKfQkDcfnP80tESCMJdCzPT7MIWwsndssF5TdaVEw9VtppHpbNRG790Wya4cyZ
lFblg5pmHGKBHQDKZBu7rmvJ//0JB/x9sC4+5J76va+BVzV+NDh94rrmPC0A6rLrvB8LDnRymIRhzIRy
N7H/s35UdeCB8TauDZ/Y1yCIV8iQB1/hdjYAQuAORM5QaKHnag8Mv1Hjzjfk/TyBn06xjg/UgOrxm95v
vmlx1HzRmNcGqynssN0lwjciaE6WUz9GWeAHaTuaT9CRYDSMvQMSOivc80K1X2qGFZw/ztBspnWwfo2
CkzNPnh9RWgl+BH0Pt9XCMR7M8QNW0cOpJFGLfypwJF/rfUreAv3XwMD0+7146z5JCM3CzFsRr1N1+qy
tQ== menoc.sk27@gmail.com
jenkins@carlos-Aspire-E1-522:~$
```

- Copie todo eso al clipboard
- Entre a github en su cuenta → Settings



- Escoja SSH and GPG Keys, agregue una nueva con el nombre que guste y copie la llave que copió de la terminal.

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

Saved replies

Authorized applications

Installed integrations

Developer settings

OAuth applications

Integrations

Personal access tokens

SSH Added on Aug 15, 2016 by GitKraken — Never used

Jenkins
Fingerprint: b1:dc:e9:9e:c2:76:e4:c4:fa:39:80:1c:9a:7d:68:51 Delete

SSH Added on Oct 14, 2016 — Never used

Title
Jenkins 2

Key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQC8Nzn6Mky/dsJ1kPl1gksjhygfasqtqwyv3loogdZ1TPWeNwgX6
NGN+Ohd3681WYHrf+8Py2PUpCiJHx/lq+Tmtlp3Ptkhu+11LK9Qi1e/mudXmtCaGV9C9lEuSzJzqLjs44fGQIN
x9pIMazak+kevmD9iDX9/69z+plKPM2U9GhTq2/gqBMQTCGRzHsJxw3BAuySOuOaaNW9ijnHhs3nbpdk3o
Eb4smhXFd5QOIC6C1f0e20K5Hfga4llzvmIsT2+St6VLxTCLwmdssb0yywR7lscZzf5HeYJogOGbYWrKfQkD
cfnP8OtESCMJdCzPT7MIWWsndssF5TdaVEw9VtpHpbNRG79Owya4cyZlFblg5pmHGKBHQDKZBu7rmvJ
/0JB/x9sC4+5J76va+BVzV+NDh94rrmPC0A6rlvB8LDnRymIRhziRyN7H/s35UdeCB8TauDZ/Y1yCIV8iQB/
hdjYAQuAORM5QaKHnag8Mv1Hjzjfk/TyBnO6xgl/UgOrxm95vwm1x1HzRmNcGqynssNolwjciaE6WUz9GWe
AHaTuaT9CRYDSMvQMSOivc8OK1X2qGFZw/zlBspnWwfo2CkzNPnh9RWgl+BH0Pt9XCMR7M8QNW0cO
pJFGlfpwJF/rfUreAv3XwMD0+7146z5JCM3CzFsRr1N1+qytQ== menoc.sk27@gmail.com

Add SSH key

Luego, dé Add SSH Key para agregarla.

- Ahora, el comando que antes dio error no lo dará. Sin embargo, esto no nos importa ya, porque cumplió su propósito de hacer que Jenkins conozca Git.

Veamos ahora cómo configurar el control de versiones en la interfaz de Jenkins. Estábamos

Enter an item name

ACS_TempSeg

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate workspace, so you can have multiple things of the same name as long as they are in different folders.

OK

aquí:

y dimos ok. La pantalla

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Project name ACS_TempSeg

Description

[Plain text] Preview

☐ Discard old builds

☒ GitHub project

Project uri https://github.com/menocsk27/ACS_SegmentTemp

☐ This project is parameterized

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds if necessary

Save Apply

que tenemos ahora es esta:

Es-

cogemos GitHub project y copiamos la dirección del repositorio, luego bajamos aquí:

Source Code Management

☐ None
☒ Git

Repositories

Repository URL:

Credentials: [Add](#)

[Advanced...](#)
[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any'):

[Add Branch](#)

Repository browser:

[Save](#) [Apply](#) [Add](#)

Tras copiar ese SSH, si algo de lo anterior no salió, tirará un error que produce náuseas y tendencias suicidas. Pero si todo salió bien, no tira error y no tira nada anormal, dé click en Save.

Jenkins ahora tiene un proyecto o Job configurado con la dirección de nuestro repositorio y está listo para ejecutar rutinas cada vez que se hace un commit. Ahora probaremos la ejecución de una rutina simple, para asegurarnos que todo salió como esperábamos.

Volvamos a la página de configuración del proyecto dando click en Configure. Bajamos hasta las últimas opciones, en la de Build, damos click en Add build step y escogemos Execute Shell. Se abre un espacio para escribir el comando, donde haremos simplemente un echo.

Build

[Execute shell](#)

Command:

[See the list of available environment variables](#)

[Add build step](#)

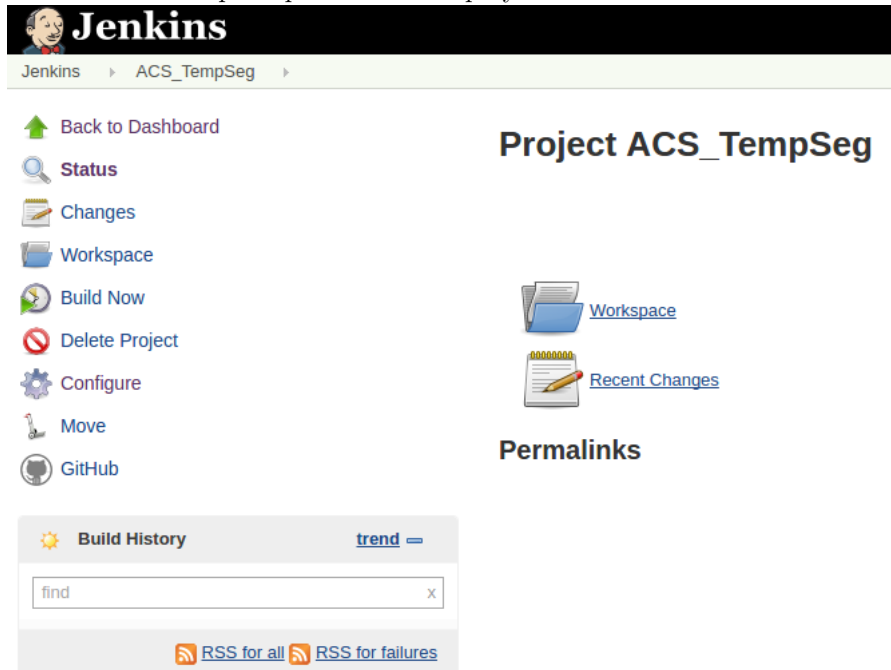
Post-build Actions

[Save](#) [Apply](#)

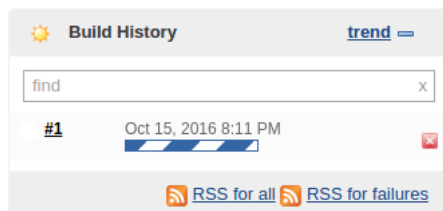
Damos Save una vez más.

5. La gloria.

En la ventana principal del Job del proyecto le damos Build Now.



Aparecerá esto:



Si damos click sobre el build y escogemos Console Output aparecerá la consola con el progreso. La primera vez tarda bastante porque hace fetch a todo el repositorio (claramente esto es dependiente al tamaño del repositorio en cuestión).



Al finalizar, veremos la gloria.

```

Started by user Julián Méndez
Building in workspace /var/lib/jenkins/workspace/ACS_TempSeg
Cloning the remote Git repository
Cloning repository git@github.com:menocsk27/ACS_SegmentTemp.git
> git init /var/lib/jenkins/workspace/ACS_TempSeg # timeout=10
Fetching upstream changes from git@github.com:menocsk27/ACS_SegmentTemp.git
> git --version # timeout=10
> git fetch --tags --progress git@github.com:menocsk27/ACS_SegmentTemp.git +refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url git@github.com:menocsk27/ACS_SegmentTemp.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url git@github.com:menocsk27/ACS_SegmentTemp.git # timeout=10
Fetching upstream changes from git@github.com:menocsk27/ACS_SegmentTemp.git
> git fetch --tags --progress git@github.com:menocsk27/ACS_SegmentTemp.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision bdf32dcd7481cc3a38a3870f0b0c382f430121f4 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f bdf32dcd7481cc3a38a3870f0b0c382f430121f4
First time build. Skipping changelog.
[ACS_TempSeg] $ /bin/sh -xe /tmp/hudson2624752669940749458.sh
+ echo Hello world, Jenkins here!!
Hello world, Jenkins here!!
Finished: SUCCESS

```

La idea es que en el comando donde de momento hay un echo puedan escribirse n pruebas unitarias que hacer con el código antes de un commit, pero de momento no tenemos pruebas unitarias de este tipo. Un ejemplo útil de algo que podemos escribir en lugar de ese echo es el comando:

```

cp * /var/www/html/ -rf
sudo /etc/init.d/apache2 restart

```

Con lo que se reinicia el server de apache cada vez.

Créditos: La información principal de cómo configurar Jenkins fue obtenida de este video <https://www.youtube.com/watch?v=3-BgaDa5B0g> y la guía fue en colaboración de los estudiantes William Astorga y Pedro Rodríguez, que son mucho más diestros en Linux de lo que yo planeo ser algún día.

4. Versión actual del sistema ¿Cómo obtenerla?

Para esto necesitamos iniciar la herramienta de Git, la cual es una aplicación CLI (*Command Line Interface*) que también suele ser usada encapsulada tras una aplicación gráfica como lo es Github Desktop, que es la herramienta utilizada por los miembros del equipo de desarrollo de este proyecto.

```
cd CarpetaCopia
git clone https://github.com/menock27/ACS_SegmentTemp.git
cd ACS_SegmentTemp
```

```
posh-git ~ ACS_SegmentTemp [master]
E:\GitBitBucket\CarpetaCopia> git clone https://github.com/menock27/ACS_SegmentTemp.git
Cloning into 'ACS_SegmentTemp'...
remote: Counting objects: 1868, done.
remote: Compressing objects: 100% (61/61), done.
remote: Total 1868 (delta 18), reused 0 (delta 0), pack-reused 1807
Receiving objects: 100% (1868/1868), 128.38 MiB | 815.00 KiB/s, done.
Resolving deltas: 100% (522/522), done.
Checking out files: 100% (1172/1172), done.
E:\GitBitBucket\CarpetaCopia> ls

Directory: E:\GitBitBucket\CarpetaCopia

Mode                LastWriteTime         Length Name
----                -
d-----          15/10/2016  11:26 p. m.         ACS_SegmentTemp

E:\GitBitBucket\CarpetaCopia> cd ACS_SegmentTemp
E:\GitBitBucket\CarpetaCopia\ACS_SegmentTemp [master =>] > ls

Directory: E:\GitBitBucket\CarpetaCopia\ACS_SegmentTemp

Mode                LastWriteTime         Length Name
----                -
d-----          15/10/2016  11:25 p. m.         Asignacion_Responsabilidades
d-----          15/10/2016  11:25 p. m.         Auditoria_13octubre
d-----          15/10/2016  11:25 p. m.         Avancel_Miscelaneo
d-----          15/10/2016  11:25 p. m.         Diseño
d-----          15/10/2016  11:25 p. m.         Herramientas
d-----          15/10/2016  11:25 p. m.         Implementación
d-----          15/10/2016  11:26 p. m.         Miscelaneo
d-----          15/10/2016  11:26 p. m.         PoC_Miscelaneo
d-----          15/10/2016  11:26 p. m.         Requerimientos
-a-----          15/10/2016  11:25 p. m.           232 .classpath
-a-----          15/10/2016  11:25 p. m.           117 .gitignore
-a-----          15/10/2016  11:25 p. m.           389 .project
-a-----          15/10/2016  11:26 p. m.           538 cortesObtenidos.csv
-a-----          15/10/2016  11:26 p. m.        36341 eclipse-java-google-style.xml
-a-----          15/10/2016  11:26 p. m.           82 README.md
-a-----          15/10/2016  11:26 p. m.        4365 tareas.txt
-a-----          15/10/2016  11:26 p. m.           94 xuvfmymb.hbl.txt

E:\GitBitBucket\CarpetaCopia\ACS_SegmentTemp [master =>] >
```

Los comandos anteriores permiten el clonado del repositorio y de su estructura general, la cual está ordenada de acuerdo al tipo de ítem de configuración de software albergados en cada carpeta. El primer comando establece la carpeta en la cual se desea alberga el repositorio, la cual llamaremos *CarpetaCopia*. Para este procedimiento, **se asume que el usuario ya creó dicha carpeta**. Una vez dentro de la carpeta, procedemos a clonar el repositorio mediante el comando *git clone* e insertando como parámetro la URL del repositorio. Este comando creará una carpeta con la estructura del repositorio tal y como se encuentra en la rama principal, la cual es llamada *master* en Git. Este copiado o clonado del repositorio copiará todos los ítems de administración de la configuración del software a la terminal local.

5. Métricas

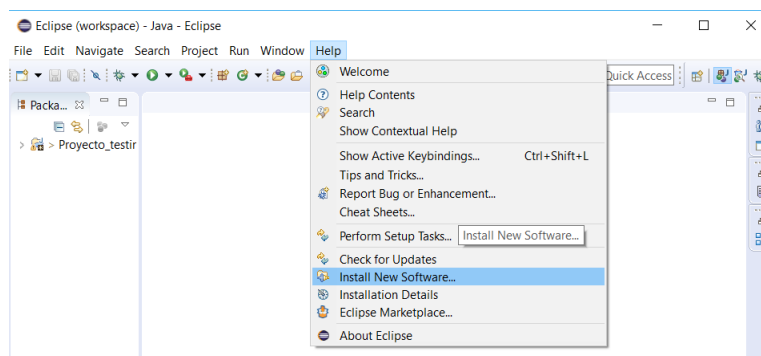
Las métricas son métodos para poder cuantificar un atributo de calidad específica de un proyecto o sistema. En el aseguramiento de la calidad del software la utilización de métricas es esencial para el mismo ya que permite mostrarle al cliente final como a demás entidades externas un valor preciso de la calidad y seguridad con la que cuenta el proyecto. En este proyecto se utilizarán distintas métricas para poder cuantificar la precisión y fiabilidad de los resultados del análisis de segmentación temporal de videos, además de qué tanta mantenibilidad se le puede dar al sistema y demás atributos de calidad externos cuantificados mediante distintas métricas, elegidas en etapas posteriores de desarrollo.

Una métrica usada en este proyecto es la cuantificación de los falsos positivos y falsos negativos obtenidos tras el análisis de segmentación temporal del video y en base a un archivo ground truth subido. Esta métrica cuantifica el atributo de precisión perteneciente a la categoría de funcionalidad, según el estándar ISO-9126 de la calidad del software. La razón de el porqué se enlazó esta métrica con dicho atributo es que la función principal del video es la correcta separación del video en escenas, por lo que la precisión de los cortes debe ser elevada. Esta métrica es cuantificada en este sprint, obteniendo los valores de 2 falsos positivos (aquellos frames que sí son interpretados como cortes pero que según el archivo ground truth, no lo son) y de 372 falsos negativos. Otras de las metricas utilizadas es la complejidad ciclomática de McCabe. Esta es usada para medir el nivel de mantenibilidad del sistema, más específicamente el atributo de analizabilidad, ya que si se lleva un valor cuantificable de qué tan complejo se ha vuelto el código fuente del sistema, se podría saber qué tan factible es que ocurran errores en etapas de mantenimiento debido a lo innecesariamente complejo del mismo. El promedio de la complejidad ciclomática en el proyecto fue de 1.3.

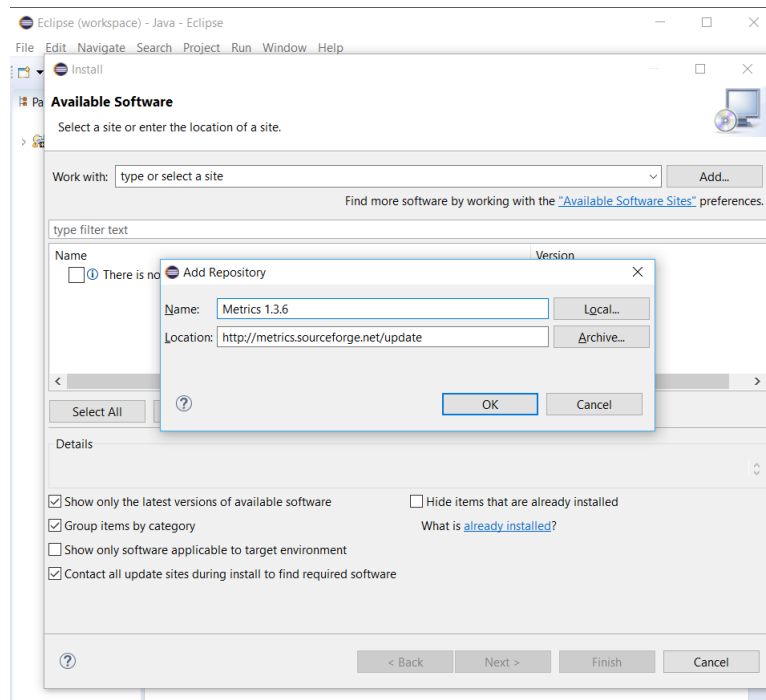
La complejidad ciclomática es medida en este proyecto utilizando el plug-in *Metrics* para Eclipse, la versión más reciente para la entrega d e este documento es la 1.3.6.

Instalación del plug-in:

- En eclipse abrir el menú de ayuda, en este seleccionar la opción de "Install New Software"



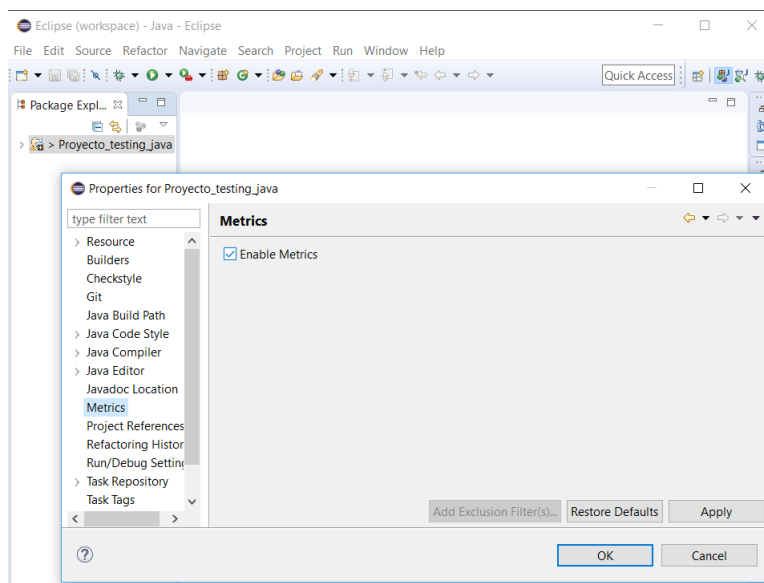
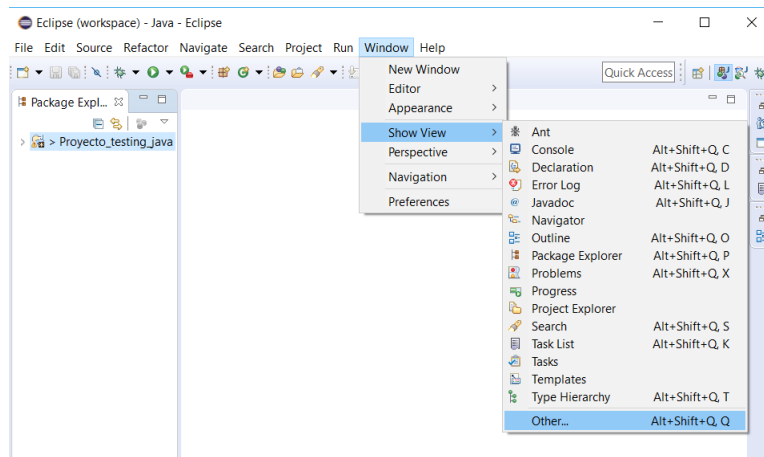
- En la ventana emergente hacer click al botón de la opción "Add..."
- Poner un nombre significativo, ej: Metrics 1.3.6, y en la opción de locación poner: `http://metrics.sourceforge.net/update`
- Presionar "Ok"



- Esperar a que debajo de "Name" aparezca una opción con una flecha que dice "Uncategorized".
- Una vez que aparezca la opción mencionada en el punto anterior marcar el cuadro blanco para que aparezca un check. Presionar el botón "Next"
- Presionar el botón de "Next", aceptar los terminos y condiciones
- Presionar el botón de "Finish"

Configurar el proyecto para permitir que el plug-in genere métricas:

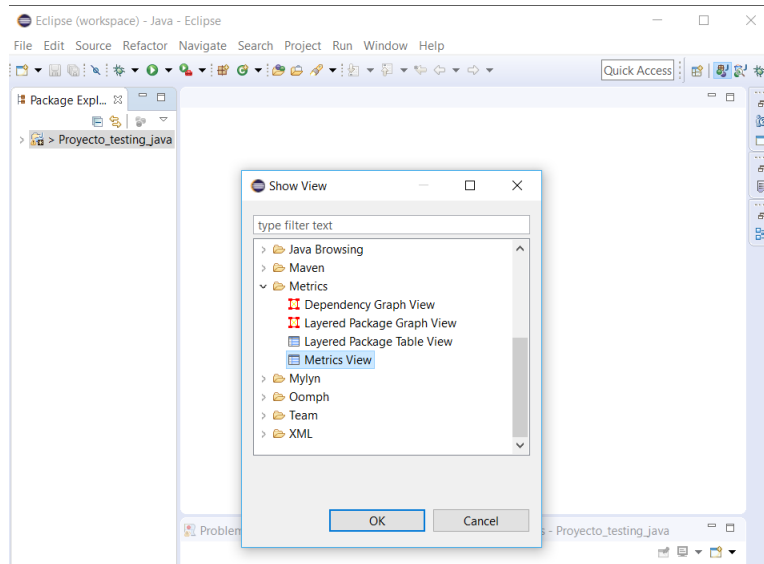
- En la ventana de paquetes de Eclipse, hacer click derecho al proyecto del que se quiere generar métricas.
- Abrir las propiedades del proyecto y navegar al menú de "Metrics"
- Marcar la opción de "Enable Metrics"
- Presionar el botón de "OK"



De esta forma se ha configurado el proyecto de forma tal que cada vez que se compile se generen las métricas que permite Metrics 1.3.6.

Para ver las métricas generadas:

- Navegar al menú de forma: Window - Show View - Other...
- Buscar la carpeta de Metrics
- Seleccionar Metric View y presionar el botón de OK



Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
▼ McCabe Cyclomatic Complexity (avg/max per method)	1.333	0.642		3	/Proyecto_testing_java/src/main/VideoSegmenter.java	split_video_to_fram...
▼ src	1.333	0.642		3	/Proyecto_testing_java/src/main/VideoSegmenter.java	split_video_to_fram...
> main	1.556	0.685		3	/Proyecto_testing_java/src/main/VideoSegmenter.java	split_video_to_fram...
▼ server	1.286	0.7		3	/Proyecto_testing_java/src/server/WebSocketBehaviour.java	onMessage
▼ WebSocketBehaviour.java	1.333	0.745		3	/Proyecto_testing_java/src/server/WebSocketBehaviour.java	onMessage
▼ WebSocketBehaviour	1.333	0.745		3	/Proyecto_testing_java/src/server/WebSocketBehaviour.java	onMessage
onMessage	3					
onClose	1					
onError	1					
onConnect	1					
sendFiles	1					
notify	1					
▼ WebSocketTest.java	1	0		1	/Proyecto_testing_java/src/server/WebSocketTest.java	main
▼ WebSocketTest	1	0		1	/Proyecto_testing_java/src/server/WebSocketTest.java	main
main	1					
> unittests	1	0		1	/Proyecto_testing_java/src/unittests/TestUnit.java	initialize

Figura 2. La imagen es un ejemplo de como se ve la Complejidad Ciclomática de McCabe dado por Metrics 1.3.6.

6. Pruebas Unitarias

En esta sección se definen un conjunto de pruebas unitarias realizadas para este proyecto. Dichas pruebas unitarias fueron realizadas utilizando el framework de testeo **JUnit**. Dicho framework permite la realización de pruebas unitarias y de integración de una manera simplificada y ordenada en el mismo entorno de desarrollo utilizado para la implementación general del proyecto, en este caso **Eclipse**. A continuación, se muestra la estructura general de cuatro pruebas unitarias implementadas para este avance. Se recuerda que una prueba unitaria es una prueba que valida la correcta funcionalidad de un método o función específica del código, en este caso y al implementarse el proyecto en lenguaje java, una prueba unitaria validará el correcto funcionamiento de un método.

■ Prueba 1

Método a evaluar: *VideoSegmenter.splitVideosToHSV()*

Entradas esperadas: Dirección local de un archivo de video mp4 o avi.

Entradas atípicas: Dirección local de un archivo no multimedia o un archivo inexistente.

Salidas: Conjunto de objetos MAT que representan los frames hsv del video ingresado mediante ruta.

■ Prueba 2

Método a evaluar: *CsvGroundtruthreader.getAbsolutedcuts()*

Entradas esperadas: Dirección local de un archivo groundtruth csv con los rangos de los cortes en el video.

Entradas atípicas: Dirección local de un archivo que no sea formato csv o inexistente.

Salidas: Conjunto de Tuplas que indican el frame inicial y final de cada corte.

■ Prueba 3

Método a evaluar: *StatisticalCalculator.calculateStdDeviation()*

Entradas esperadas: Conjunto no vacío de valores decimales.

Entradas atípicas: Conjunto vacío de valores decimales o conjunto

Salidas: Valor flotante que indica la desviación estándar del conjunto ingresado como parámetro.

■ Prueba 4

Método a evaluar: *StatisticalCalculator.calculateAverage()*

Entradas esperadas: Conjunto no vacío de valores decimales.

Entradas atípicas: Conjunto vacío de valores decimales o conjunto

Salidas: Valor flotante que indica la media del conjunto ingresado como parámetro.

Referencias

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

DevOps Library *Using Source Control in Jenkins: DevOps Library Jenkins*

Obtenido de: <https://www.youtube.com/watch?v=3-BgaDa5B0g> el 10 de octubre de 2016.