

Instituto Tecnológico de Costa Rica

Avance 2, proyecto semestral

Michael Choque, 201501858
Nelson Gómez, 2013389553
William Espinoza Ramirez, 2013100622

Contents

1 Documento SyRS	3
1.1 Propósito del sistema	3
1.2 Alcance del sistema	3
1.3 System overview	3
1.3.1 Contexto del sistema	3
1.3.2 Funciones del sistema	3
1.3.3 Características de usuario	4
1.3.4 Requerimientos funcionales	4
1.3.5 Requisitos de usabilidad	5
1.3.6 Requerimientos de rendimiento	5
1.3.7 Interfaces de sistema	5
1.4 Operaciones del sistema	5
1.4.1 Requerimientos de integración del sistema humano	5
1.4.2 Mantenibilidad	6
1.4.3 Confiabilidad	6
1.4.4 Modos de sistema y estados	6
1.5 Características físicas	6
1.5.1 Requerimientos físicos	6
1.5.2 Requerimientos de adaptabilidad	6
1.5.3 Condiciones ambientales	6
1.5.4 Seguridad del sistema	7
1.5.5 Administración de la información	7
1.5.6 Políticas y regulaciones	7

1.5.7	Mantenimiento del ciclo de vida del sistema	7
1.5.8	Embalaje, manipulación y transporte	8
1.5.9	Verificación	8
1.5.10	Suposiciones y dependencias	10
2	Estándar de Codificación	11
2.1	Utilización de Pylint	11
3	Diagramas	15
3.1	Diagrama de componentes	15
3.2	Diagrama de clases	16
3.3	Diagrama de deployment	16
4	Validación del diseño	16
5	Especificación pruebas unitarias	17
5.1	test_cargar_imagenes	17
5.2	test_vectorizar_imagen	18
5.3	test_matriz_de_imagenes	18
5.4	test_matriz_de_covarianza	18
5.5	test_auto_valores_vectores	18
5.6	test_pesos	19
5.7	test_clasificar	19
5.8	test_carga_de_entrenamiento	19
6	Manual de Administración de la Configuración del Software	20
6.1	Instalación de Pip	20
6.2	Instalación de Flask	21
6.3	Instalación de Github Desktop y Clonación del Repositorio	21
7	Actividades	23
8	Métricas implementadas	23
8.1	Eficiencia de carga de los archivos	23
8.2	Métrica de precisión	24

1 Documento SyRS

1.1 Propósito del sistema

Realizar un sistema de reconocimiento facial para implementar lo que se conoce como pasaporte electrónico. Con el fin de que se agilicen los procesos de identificación de personas en los aeropuertos y evitar el uso de papeles en el aeropuerto para que el medio ambiente se vea beneficiado. Dicho sistema deberá seguir ciertos atributos de calidad y métricas las cuales garantizarán al usuario la satisfacción necesaria.

1.2 Alcance del sistema

Los puntos de control del país buscan un proceso ágil en el que se puedan reconocer a los sujetos que entran y salen del país. Para suplir dicha necesidad, se se implementará un sistema de reconocimiento facial capaz de identificar a todas las personas que ingresen y dejen el país. Tal sistema deberá ser preciso y determinar de forma inequívoca a alguna persona. Además, se espera que el sistema haga dichas labores con alta velocidad, con el fin de reducir la espera de los sujetos en los puntos de control. El sistema proveerá una sesión de entrenamiento en el cual se registrarán a las personas que podrán ser identificadas. Cabe destacar que el sistema únicamente será para reconocimiento de sujetos, no incluirá identificación de reos ni determinará si una persona está siendo seguida por la interpol.

1.3 System overview

1.3.1 Contexto del sistema

Este documento posee una descripción de alto nivel del sistema de reconocimiento de rostros que será implementado en los puestos de aduanas de los aeropuertos internacionales de Costa Rica. La inspiración para el presente proyecto nace dada la necesidad de expeditar el proceso de ingreso a Costa Rica sin comprometer la seguridad del aeropuerto, los usuarios y empleados del mismo y a los ciudadanos del país. Esto se planea realizar al utilizar el reconocimiento de rostros, tanto para lo ya mencionado como también para verificar las caras de sujetos buscados internacionalmente, al comparar las imágenes capturadas por el sistema contra una base de datos de rostros pertenecientes a sospechosos buscados por distintos crímenes, un proceso mucho más eficiente que las comparaciones manuales realizadas por los empleados de aduana.

1.3.2 Funciones del sistema

El sistema se encontrará dividido en dos módulos, el de entrenamiento y el de reconocimiento.

Entrenamiento El sistema podrá ser entrenado para que reconozca sujetos, para esto se deberá de pasar al sistema los archivos con al menos 10 imágenes por sujeto, con lo cual se podrá realizar el entrenamiento y más adelante el reconocimiento de una persona. El sistema deberá realizar el entrenamiento antes de identificar sujetos, en caso contrario simplemente no se identificara a una persona. Este módulo será utilizado por operadores, dichos operadores se registraran de manera directa a la base de datos (no habrá un registro en el sistema).

Según las necesidades prioritarias para el modulo de entrenamiento se tendrá:

- Carga de un conjunto de imágenes almacenada en la dirección provista por el usuario para el entrenamiento. El conjunto de imágenes provisto debe contener una carpeta por cada sujeto,

y se debe almacenar el nombre de tal carpeta como la etiqueta de tal sujeto o clase.

- Entrenar el sistema, generando las auto-caras y la proyección de las muestras en el nuevo espacio formado por tales auto-caras, usando el algoritmo supervisado del centroide más cercano. Tanto las auto-caras como las proyecciones deben ser almacenadas usando algún esquema de almacenamiento de su elección. Se tendrá de manera parametrizable el porcentaje de auto-vectores a conservar en el nuevo conjunto de vectores base.
- Cargar un conjunto de muestras de pruebas, para medir la precisión del sistema.

Reconocimiento El sistema podrá reconocer si cierto sujeto es quien dice ser por medio de la información que se haya obtenido en el módulo de entrenamiento. No se podrá saber información extra del sujeto además de la siguiente: nombre, cédula, estado civil, nacionalidad, fecha de nacimiento.

En este avance la sección de reconocimiento se verá únicamente como un sujeto con sus respectivas imágenes será reconocido pero no se mostrarán detalles de su información.

1.3.3 Características de usuario

En el segundo sprint del proyecto, el sistema de reconocimiento facial deberá poder ser usado por un usuario únicamente. El cual logre realizar el entrenamiento de los sujetos. El usuario deberá proporcionar la dirección en la que se encuentran las imágenes que alimentará al sistema.

1.3.4 Requerimientos funcionales

- Registro de nuevos usuarios al sistema, se le deberá de pasar al sistema tanto al menos 10 imágenes del nuevo usuario como la información básica del mismo (nombre, cédula, estado civil, nacionalidad, fecha de nacimiento).

Para este sprint lo que se realizara será pasarle al sistema una carpeta que contenga dentro de esta varias carpetas cuyos nombres representaran a un nuevo sujeto por agregar al sistema, cada carpeta del sujeto tendrá a su vez dentro de sí mismos las imágenes que representen a dicho sujeto.

Prioridad: Baja

- Entrenamiento de varios sujetos de una vez, se deberá pasar al sistema la ruta con donde se encuentren los sujetos y sus imágenes en diferentes carpetas.

Prioridad: Alta

- Reconocimiento de un usuario, el usuario deberá de usar la cámara para que el sistema saque una foto de su rostro y pueda iniciar el reconocimiento del mismo.

Para este sprint el reconocimiento se realizara dando al sistema la dirección donde se encuentre la imagen del sujeto a reconocer.

Prioridad: Media

- Registro de usuarios operadores, podrán utilizar el módulo de entrenamiento para ya sea ingresar nuevos usuarios de uno en uno o varios de una vez.

Como ha sido mencionado anteriormente el sistema en este avance no posea una diferenciación entre un usuario operador y un usuario común.

Prioridad: Baja

- Login de usuarios operadores, el sistema identificará un operador según un nombre de usuario y una contraseña.

El login no será realizado en este sprint para poder concentrarse en las necesidades prioritarias.

Prioridad: Baja

1.3.5 Requisitos de usabilidad

El sistema deberá ser veloz y preciso en la sección de identificación de individuos. En la parte de entrenamiento se deberán proveer 10 imágenes del sujeto. Y se espera que sea lo más intuitivo posible para las personas que se identificarán en él, es decir, un sistema agradable para el usuario y de fácil uso.

1.3.6 Requerimientos de rendimiento

- Se espera llegar a soportar información de al menos todas las personas en Costa Rica. Para este sprint se soportara la informacion de una pequeña base de datos de 41 sujetos.
- En cuanto a lo que es la cantidad de usuario que se puedan conectar a la vez, se medirá lo que el sistema aguante con lo que luego se podrá estimar la cantidad de terminales a instalar en los aeropuertos.
- Se espera que la frecuencia de uso del sistema sea muy alto, por lo que este deberá de poder soportar un uso constante.

1.3.7 Interfaces de sistema

Las interfaces deben ser lo más fáciles posibles para los usuarios, ya que de ello dependerá que tan rápido los usuarios logren utilizar el sistema, con lo que se podrán reducir o no las colas de espera en el aeropuerto.

La interfaz de este sprint no sera tan amigable con el usuario por cuestiones de seguridad con los browser, el usuario debera de meter la direccion donde se encuentren los sujetos de manera manual.

1.4 Operaciones del sistema

1.4.1 Requerimientos de integración del sistema humano

Durante este según sprint, el sistema será utilizado por unicamente un usuario, por lo cual, lo unico importante es que el usuario proporcione la dirección de las imagenes de los sujetos para relizar el entrenamiento. Sin embargo, una vez terminadas todas las funcionalidades del sistema y se ponga en marcha el sistema, los requerimientos de Integración del sistema humano se dividen en las siguientes categorias:

- Personal de trabajo: Para el funcionamiento del sistema será necesaria la inclusión de un equipo que pueda proveer el mantenimiento necesario a la plataforma física utilizada para el reconocimiento de rostros. Además de personal encargado del control del uso del sistema para asegurar tanto que el sistema está siendo utilizado de manera correcta como también para verificar que las personas estén siguiendo un orden adecuado.
- Entrenamiento: Se observa como una necesidad el entrenamiento del personal de seguridad y de aduanas para permitir la correcta operación e interacción con el sistema de reconocimiento de rostros y las alertas levantadas por el mismo. Malentendidos respecto a la operación de la plataforma física o digital del sistema podrían generar retrasos en el proceso de aduanas e incluso acabar en la no utilización del sistema debido a la poca o ninguna capacitación del personal.
- Seguridad: Se requerirá de un personal encargado de escoltar a todas aquellas persona identificadas como buscadas por agentes policiaos y también para aquellos sujetos cuyo reconocimiento haya resultado como erróneo. Esto con el fin de procurar un ambiente de mayor confianza y seguridad entre los usuarios del sistema y el aeropuerto.

- Factores humanos ingenieriles: El sistema deberá de proveer de alguna herramienta que ayude al usuario a entender cómo se usa el software de manera tal que lo guíe entre lo que debe hacer para una mejor comprensión y un rápido aprendizaje del sistema.

1.4.2 Mantenibilidad

La mantenibilidad en un sistema de esta magnitud es algo muy importante a considerar debido al ámbito en el que se pondrá en funcionamiento. La estabilidad debe ser un factor primario, los tiempos de inactividad del sistema deben ser reducidos al mínimo ya que un fallo en el mismo podría generar inconvenientes en las personas que necesitan tomar vuelo. La cambiabilidad debe ser un asunto que se puede combatir con el nivel modularidad de cómo esté desarrollado el sistema, ya que no puede estar inactivo durante mucho tiempo, se necesita que mientras se cambie o integre alguna parte del sistema, el resto esté funcionando con normalidad.

1.4.3 Confiabilidad

Los requerimientos de confiabilidad serán cumplidos siempre y cuando se logre una alta tolerancia a los fallos, y alta recuperación en caso que ocurra algún inconveniente. Ningún sistema está hecho para que falle, pero en caso que ocurra, la recuperación en el menor tiempo posible debe ser muy importante. Además es importante que una vez que ocurra un fallo no hayan pérdidas de datos.

1.4.4 Modos de sistema y estados

El sistema tendrá 2 modos distintos: El modo de reconocimiento de sujetos y el modo de entrenamiento. En el modo de reconocimiento se deberá utilizar una cámara para capturar el rostro de la persona e iniciar el proceso de identificación del mismo. En el modo de entrenamiento se deberán proveer las imágenes del sujeto que serán guardadas en el sistema, mediante una ruta de la computadora.

1.5 Características físicas

1.5.1 Requerimientos físicos

Se contará con varios servidores primarios con replicación en diferentes puntos del país en modo de Cluster. y diferentes extensiones en los puntos de entrada del país los cuales son, los 2 aeropuertos, Juan santamaría y tobías bolaños, y las vías terrestres, paso canoas, sixaola y peñas blancas.

1.5.2 Requerimientos de adaptabilidad

El sistema será desarrollado con un pensamiento a futuro, deberá ser lo más modular posible, y además, con alta adaptabilidad a otros sistemas. Ya que se necesitará integrar sistema de búsqueda de personas a nivel internacional o sistemas del FBI etc. Los cambios que se originen y el tiempo de inactividad de ciertas partes deben ser mínimos durante el cambio en el software.

1.5.3 Condiciones ambientales

En cuanto a las condiciones en las que se debe encontrar el sistema, lo más importante es la cantidad de luz que se llegue a recibir, esto puesto que se requiere que el sistema pueda tomar una foto clara del sujeto para posteriormente identificarlo.

1.5.4 Seguridad del sistema

El sistema tendrá una sección de logueo especial para las personas que registrarán personas al sistema. Los usuarios normales no podrán acceder a dicha funcionalidad del sistema, pues está sección es únicamente para entes encargados en incluir las personas en el sistema. Otra técnica utilizada para proteger la información del sistema será la encriptación de los datos. También, la base datos contará con una bitácora para tener un registro de las personas que actúan en ella. Así en caso de fraude o cambios se tendrá un registro de la persona que lo efectuó.

1.5.5 Administración de la información

Principalmente la información que se guardara en el sistema, son las fotos de los sujetos y la información de cada sujeto, más detalladamente las fotos se manejan por medio de arreglos con vectores que contengan la información de cada pixel de las imágenes, los datos de cada sujeto se maneja con tipos primitivos de datos, aunque al estar usando python para la implementación esto no es tan relevante. La información mencionada será manejada tanto de manera local como por medio de una base de datos.

1.5.6 Políticas y regulaciones

Entre las regulaciones a tomar en cuenta para el sistema tenemos: soporte de multi lenguaje, principalmente español e inglés. Proteger la información personal de los sujetos registrados, esta se puede medir con la exactitud del sistema al tratar de reconocer a un sujeto. Reportar el uso y el acceso que ha otorgado el sistema sobre los sujetos que lo han utilizado.

1.5.7 Mantenimiento del ciclo de vida del sistema

Se considera que este sistema se encontrará en operación durante un lapso prolongado, por lo que el soporte, tanto a nivel de plataforma física como digital, es de especial importancia. Por lo tanto se consideran necesarios los siguientes puntos:

- Documentación externa e interna apropiadas para el soporte continuo del sistema. Si bien la mayor parte del soporte del sistema será proveído por la empresa que suministra este software, la colaboración con equipos externos a la misma podría ser necesaria y en dicho caso, una correcta y suficiente documentación facilita el proceso de capacitación de dicho equipo externo.
- Implementación de estándares de codificación. Debido a la eventual rotación del personal dentro de la empresa encargada de suministrar el presente software y la posible participación con equipos externos a la misma, se considera una necesidad la estandarización del código utilizado en la elaboración del proyecto. Esto debido a la facilidad que un estándar de codificación otorga al relegar el sistema a un nuevo equipo y la comprensión del código por parte de un equipo externo en caso de que sea necesaria dicha colaboración.
- Suministro de refacciones. El presente sistema requiere de una plataforma física la cual será utilizada por una cantidad considerable de personas de distintas composiciones físicas. Esto aplicará estrés y desgaste importante sobre los componentes físicos del sistema, por lo que se considera como una necesidad un proveedor que pueda suministrar las refacciones necesarias para el mantenimiento de la plataforma física del sistema.
- El uso de diagramas para facilitar la comprensión del sistema por parte de entidades externas que lleguen a realizar alguna clase de mantenimiento hacia el sistema, también para la exploración de posibles cambios o adiciones a lo que ya se tiene actualmente.

1.5.8 Embalaje, manipulación y transporte

El sistema se encontrará distribuido en distintos servidores con replicación, los cuales serán accedidos desde clientes web, por lo que no será necesario el empaquetamiento o instalación de materiales extra por parte del aeropuerto, pero sí será necesario que exista alguna clase de cubículo donde el sujeto pueda realizar la identificación.

1.5.9 Verificación

Requisitos funcionales

Registro de nuevos usuarios al sistema

Registro de nuevos sujetos sin toda la información requerida. El sistema deberá comprobar el formato, tipo de los datos ingresados y la cantidad de imágenes asociadas al nuevo sujeto.

Prioridad: Alta

Entrenamiento de varios sujetos de una vez

El sistema deberá tolerar el ingreso de al menos 41 nuevos sujetos simultáneamente.

Prioridad: Alta

Administración de usuarios operadores

El sistema deberá permitir el registro y eliminación de usuarios operadores, además de asegurar la seguridad de sus contraseñas.

Para asegurar el cumplimiento de estos se utilizara:

- Un cálculo de la cantidad de registros fallidos encontrados sobre la cantidad total de registros realizados.
- Uso de la herramienta Selenium.

Prioridad: Baja

Reconocimiento de un usuario

El sistema deberá de ser preciso y eficiente con la respuesta hacia el usuario.

Para asegurar el cumplimiento de esto se utilizara:

- Uso de la herramienta Selenium para tomar el tiempo.
- Un cálculo de la cantidad de registros fallidos encontrados sobre la cantidad total de registros realizados.
- Uso de un decorator al método de carga de imagenes para medir el tiempo de IO.
- Javascript para calcular si el usuario pudo identificar que la interfaz asociada a este método fuera comprensible.

Prioridad: Media

Login de usuarios operadores

El sistema deberá de ser seguro, garantizando la entrada a funciones de registro a aquellas personas que tengan el nivel requerido de acceso al sistema.

Para asegurar el cumplimiento de esto se utilizara:

- Encriptamiento de la información referentes a contraseñas sobre aquellos usuarios que son operadores.

Prioridad: Baja

Requisitos de usabilidad

El sistema deberá de ofrecer facilidad para que los usuarios finales aprendan el uso de la herramienta como cambio al típico uso de pasaportes para la identificación de su persona.

Para asegurar el cumplimiento de estos se utilizara:

- TimeMe.js para medir el tiempo que le toma a un usuario el aprender a utilizar una función de manera correcta.

Prioridad: Baja

El sistema deberá buscar la automatización y agilización del actual proceso de identificación de sujetos, los usuarios deberán encontrar cómodo y fácil el poder usarlo.

Para asegurar el cumplimiento de estos se utilizara:

- TimeMe.js para medir el tiempo que le toma a un usuario completar cierta funcionalidad.

Prioridad: Baja

El sistema deberá garantizar que los usuarios sean capaces de utilizar el software sin ninguna o poca dificultad, dado que la implementación de este sistema busca agilizar el proceso de ingreso al país.

Para asegurar el cumplimiento de estos se utilizara:

- Javascript para detectar qué funciones han sido utilizadas. 0 clicks indican una función no identificada
- Time.js se utilizará con el fin de detectar el tiempo que un usuario del sistema consuma en la comprensión de las funcionalidades del sistema. Este módulo permite capturar el tiempo real que se ha dedicado a una tarea, dado que el contador se detiene si la ventana actual pierde enfoque.

Prioridad: Baja

Requerimientos de rendimiento

Soportar información de todas las personas en Costa Rica

Se deberá asegurar que se tiene con los suficientes recursos para controlar toda la memoria que será necesario guardar.

Para este sprint se utilizaran solo 41 sujetos.

Prioridad: Baja

Cantidad de usuarios conectados a la vez

Se busca asegurar que el sistema sea robusto aun cuando la máxima cantidad posibles de usuarios posibles estén usándolo en un mismo momento.

Soportar uso constante

El sistema deberá aguantar al menos 1000 usuarios a la vez

Para asegurar estos puntos se utilizará:

- Herramientas de medición de tiempos sobre lo que duran los algoritmos de reconocimiento y entrenamiento.
- Selenium.
- Hacer uso de un data warehouse para guardar todos los datos de las personas.

Prioridad: Baja

Requerimientos de mantenibilidad

Para la medición de estos se utilizara:

- Un cálculo sobre la cantidad de veces en las que falla el sistema contra la cantidad de funciones que sean realizadas.
- Uso de la herramienta EclEmma para ver cuanto codigo se ejecuta y estimar cuánto costaría cambiarlo.
- Documentar las tareas realizadas sobre el sistema, con el uso de la herramienta zoho.
- Realizar la documentación respectiva de los puntos aquí mencionados en el sistema, para poder garantizar una buena mantenibilidad.

Prioridad: Media

Requerimientos de confiabilidad

Para la medición de estos se utilizará:

- Uso de las herramientas Selenium/Logging para el cálculo de la tolerancia del sistema a fallos.
- Realizar una documentación cuando los problemas sean errores manuales (desconectar cable de red, cerrar navegador al subir una imagen, interrumpir el entrenamiento del sistema)
- Sonarqube para calcular la madurez del sistema sobre los fallos.

Prioridad: Media

1.5.10 Suposiciones y dependencias

Para el uso correcto del sistema se deberá contar con una cámara. Ya que que en la sección de identificación de sujeto se deberá tomar una foto del sujeto y realizar las operaciones necesarias para el reconocimiento. También, en el modo de entrenamiento se deberán proveer 10 imágenes de la persona, esto con el fin de quedar guardado en el sistema.

2 Estándar de Codificación

Para la escritura del código en Python se utilizará el estándar PEP-8, según se define en el sitio oficial www.python.org.

Se elige este estándar de codificación debido a los siguientes atributos:

- Claridad del código: PEP-8 incluye reglas y sugerencias orientadas a limpiar y normalizar la estructura estática del código, con el fin de facilitar su lectura. La regulación de espacios e indentación facilitan la identificación de bloques de código y estructuras de control, de qué código está contenido dentro de sus límites y cuál es el flujo de ejecución del programa.
- Consistencia en el nombramiento de los elementos del código: PEP-8 incluye reglas y sugerencias orientadas a estandarizar y darle consistencia a los nombres de los distintos elementos de un programa (variables, funciones, etc). Al seguir estas reglas, facilita al autor del programa y a otros individuos que interactúan con el código, la identificación de un elemento dentro y su función, en el caso de funciones o procedimientos, o contenido, en el caso de variables, solamente al leer su nombre.
- Optimización del código: PEP-8 incluye reglas y sugerencias orientadas a aprovechar decisiones de implementación del lenguaje Python, con el fin de optimizar su ejecución y mejorar el flujo de ejecución del programa. Adicionalmente, presenta sugerencias acerca del uso apropiado de las excepciones y el manejo de errores, con el objetivo de extraer la mayor cantidad de información necesaria para determinar la cadena de eventos que llevaron a la manifestación de un fallo.

Para hacer cumplir el uso del estándar PEP-8 se utilizará la herramienta Pylint. Además de revisar el cumplimiento del estándar, permite la configuración de la convenciones y reglas que se consideren necesarias, detección de errores e integración con el ambiente de programación Eclipse.

Se utilizará el estilo de codificación Google Style para los segmentos de código Javascript. Esto debido a que cuenta con una herramienta para comprobar su implementación : clang-format.

2.1 Utilización de Pylint

Para la verificación del estándar de codificación PEP-8 en los segmentos de código escritos en el lenguaje Python se utilizó la herramienta Pylint.

A continuación se detalla el proceso para utilizar la herramienta de verificación de código Pylint en Eclipse.

La versión de Eclipse sobre la que se utilizó Pylint es Eclipse Jee Oxygen sobre el sistema operativo Windows 10, con Python versión 2.7.14.

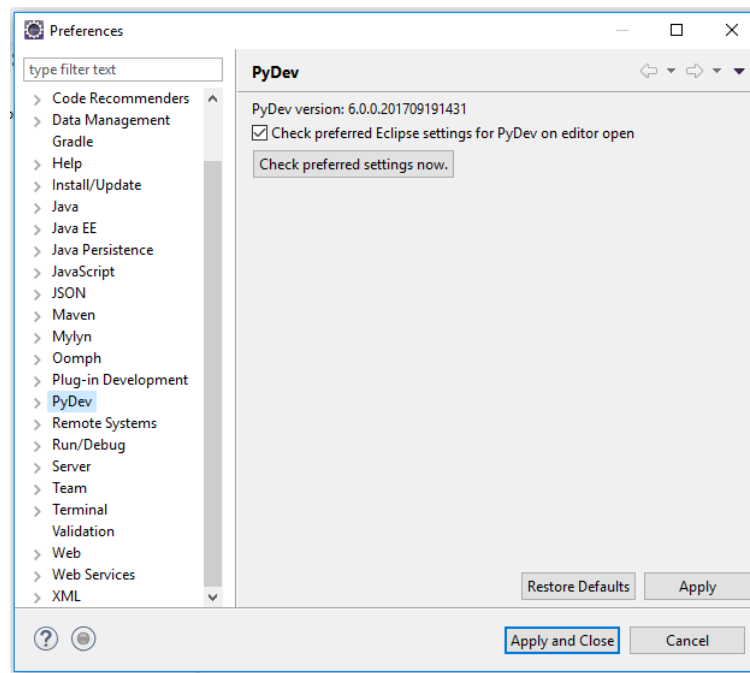
El primer paso para instalar Pylint es adquirir la herramienta de instalación Pip. Una vez adquirida, se debe abrir la consola e ingresar el comando `pip install pylint`, lo que inicia el proceso de descarga e instalación de Pylint:

```
Símbolo del sistema
C:\Users\DrkSprtn>pip install pylint
Collecting pylint
  Using cached pylint-1.7.4-py2.py3-none-any.whl
Collecting astroid>=1.5.1 (from pylint)
  Using cached astroid-1.5.3-py2.py3-none-any.whl
Collecting singledispatch; python_version < "3.4" (from pylint)
  Using cached singledispatch-3.4.0.3-py2.py3-none-any.whl
Collecting mccabe (from pylint)
  Using cached mccabe-0.6.1-py2.py3-none-any.whl
Collecting configparser; python_version == "2.7" (from pylint)
Collecting backports.functools-lru-cache; python_version == "2.7" (from pylint)
  Using cached backports.functools_lru_cache-1.4-py2.py3-none-any.whl
Collecting six (from pylint)
  Using cached six-1.11.0-py2.py3-none-any.whl
Collecting colorama; sys_platform == "win32" (from pylint)
  Using cached colorama-0.3.9-py2.py3-none-any.whl
Collecting isort>=4.2.5 (from pylint)
  Using cached isort-4.2.15-py2.py3-none-any.whl
Collecting wrapt (from astroid>=1.5.1->pylint)
Collecting enum34>=1.1.3; python_version < "3.4" (from astroid>=1.5.1->pylint)
  Using cached enum34-1.1.6-py2.py3-none-any.whl
Collecting lazy-object-proxy (from astroid>=1.5.1->pylint)
  Using cached lazy_object_proxy-1.3.1-cp27-cp27m-win32.whl
Installing collected packages: wrapt, enum34, six, singledispatch, lazy-object-proxy, backports.functools-lru-cache, astroid, mccabe, configparser, colorama, isort, pylint
Successfully installed astroid-1.5.3 backports.functools-lru-cache-1.4 colorama-0.3.9 configparser-3.5.0 enum34-1.1.6 isort-4.2.15 lazy-object-proxy-1.3.1 mccabe-0.6.1 pylint-1.7.4 singledispatch-3.4.0.3 six-1.11.0 wrapt-1.10.11
C:\Users\DrkSprtn>
```

Pylint es instalado en la dirección /Lib/site-packages dentro de la carpeta en la que Python es instalado por defecto.

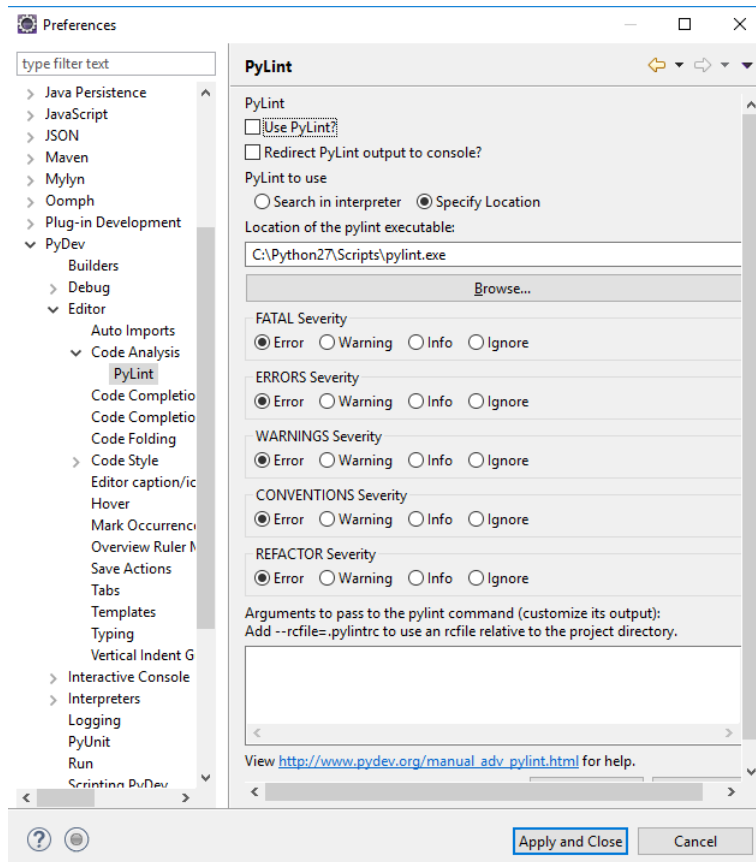
Para habilitar la verificación de código mediante Pylint se deben seguir los siguientes pasos:

1. Una vez abierto el IDE Eclipse, seleccionar Window y luego Preferences. Al abrirse esta ventana navegar a la sección dedicada a PyDev:



2. Clicar la flecha para mostrar las subsecciones. Una vez desplegadas, clicar la flecha al lado de

Editor y lo mismo para Code Analysis. Una vez aquí, clicar la sección Pylint:



3. La primera opción, Use Pylint?, activa el complemento. La segunda opción permite desviar los mensajes producidos por Pylint hacia la consola, lo que es preferible al momento de aprender el funcionamiento de la herramienta. Debajo de estas dos opciones se especifica dónde se debe buscar el ejecutable de Pylint. La segunda opción, Specify Location nos permite definir con seguridad dónde debe Eclipse buscar el ejecutable de la herramienta Pylint.

Como se observa en la imagen, el ejecutable se encuentra en el directorio de instalación de Python, dónde fue previamente instalado. 4. Las siguientes cinco secciones indican los tipos de fallas que Pylint es capaz de detectar y cómo deberían ser reportadas. La sección relevante para el análisis de código es CONVENTIONS severity. Existen 4 niveles de alarma: 1. Error: Impide que el código sea ejecutado si existen violaciones al estándar PEP-8

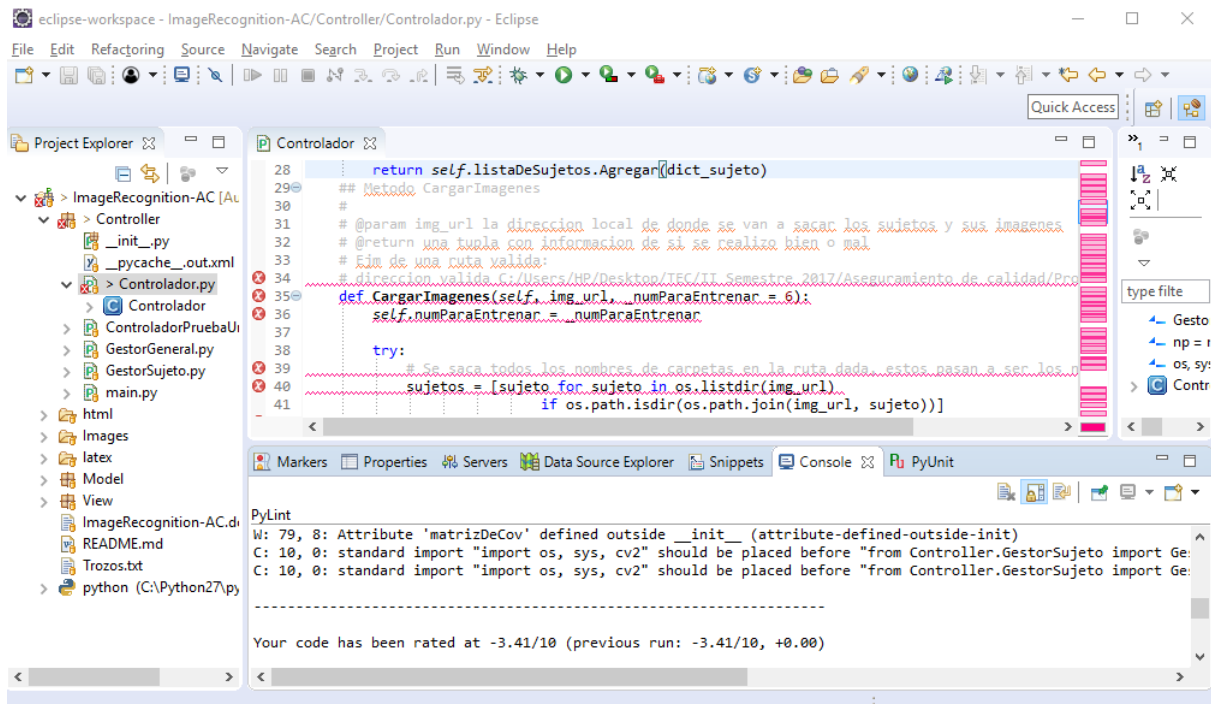
2. Warning: Se marca dónde ocurre una violación al estándar pero el código puede ser ejecutado con normalidad.

3. Info: Solamente se provee información acerca de la violación al estándar, pero no se le asigna ninguna marca especial. 4. Ignore: Violaciones al estándar serán ignoradas por completo. En este caso se utilizará la opción de Error, para evitar que el código llegue al repositorio con violaciones al estándar. Una vez establecida la configuración, clicar Apply and Close y Pylint está listo para ser utilizado.

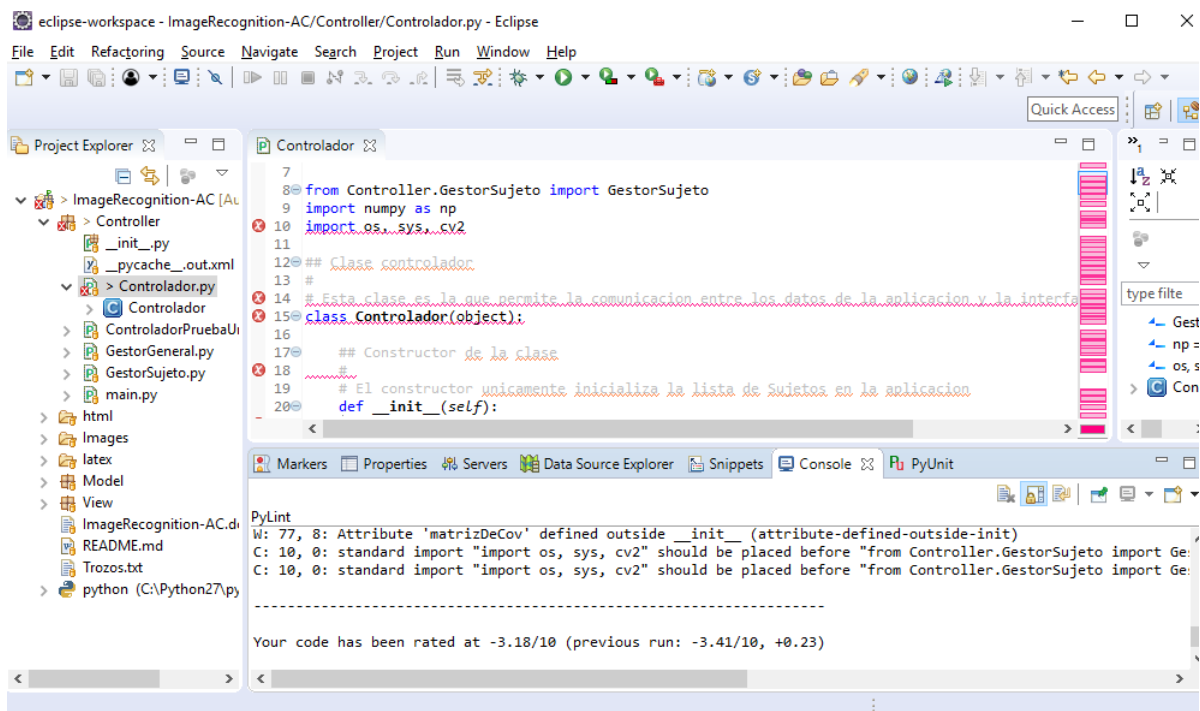
Existen 2 maneras de iniciar la verificación de código. Una se ejecuta de manera automática al guardar un archivo al cual se le han realizado cambios. Pylint realizará un escaneo del código para determinar si se han realizado violaciones al estándar PEP-8.

Para realizar una revisión sobre un archivo al cual no se le han realizado cambios primero hacer click derecho sobre el archivo a analizar, seleccionar la opción PyDev y clicar sobre Code analysis. Esto iniciará el análisis del archivo seleccionado.

Al finalizar el análisis se obtiene el siguiente resultado:



Cada marca corresponde a una violación del estándar PEP-8. En consola se especifica la violación cometida y en qué línea del archivo se encuentra. Según la proporción de violaciones y el tipo de estas, se le asigna una calificación que varía entre -10 y 10. Al lado de la calificación obtenida se muestra el incremento en la calificación desde la última comprobación de código:



En este caso se corrigieron unas cuantas violaciones al estándar y se logró incrementar ligeramente la calificación del código.

Sin embargo, existen casos en los que corregir una violación al estándar puede verse como innecesario, una molestia e incluso falsos positivos como en el caso de las librerías OpenCV, donde Pylint indica que no existen miembros dentro del módulo que sí están presentes.

Para resolver estas excepciones se pueden incluir como comentarios instrucciones que le indican a Pylint que debe ignorar las violaciones que pertenezcan al tipo especificado.

```

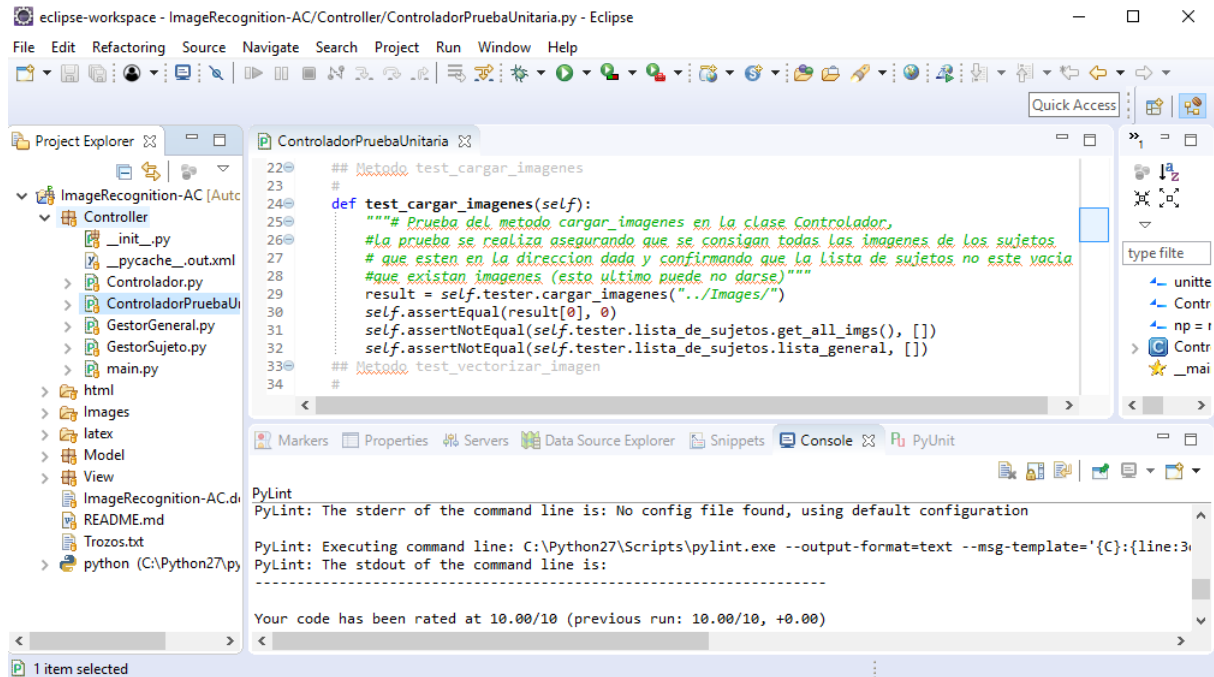
matriz_img = np.matmul([ [ 0.0, 0.0, 0.0, 0.0 ],
                          [ 0.0, 0.0, 0.0, 0.0 ] ],
                        [ [ 0.0, 0.0, 0.0, 0.0 ],
                          [ 0.0, 0.0, 0.0, 0.0 ] ])
# pylint: disable-msg=C0301
result_auto_valores, result_auto_vectores = self.testeer.definir_auto_valores_vectores(matriz_cov, matriz_img)
self.assertEqual(result_auto_valores.tolist(), [0.0, 0.0])

```

En este caso la instrucción indica que se debe ignorar la violación de la cantidad máxima de caracteres en una línea.

En la siguiente página se encuentra una lista completa de todos los códigos asociados a las distintas violaciones que Pylint puede detectar: <http://pylint-messages.wikidot.com/all-codes>

Al corregir todas las violaciones al estándar se le otorga al código una calificación perfecta que demuestra que el código analizado se conforma al estándar PEP-8:

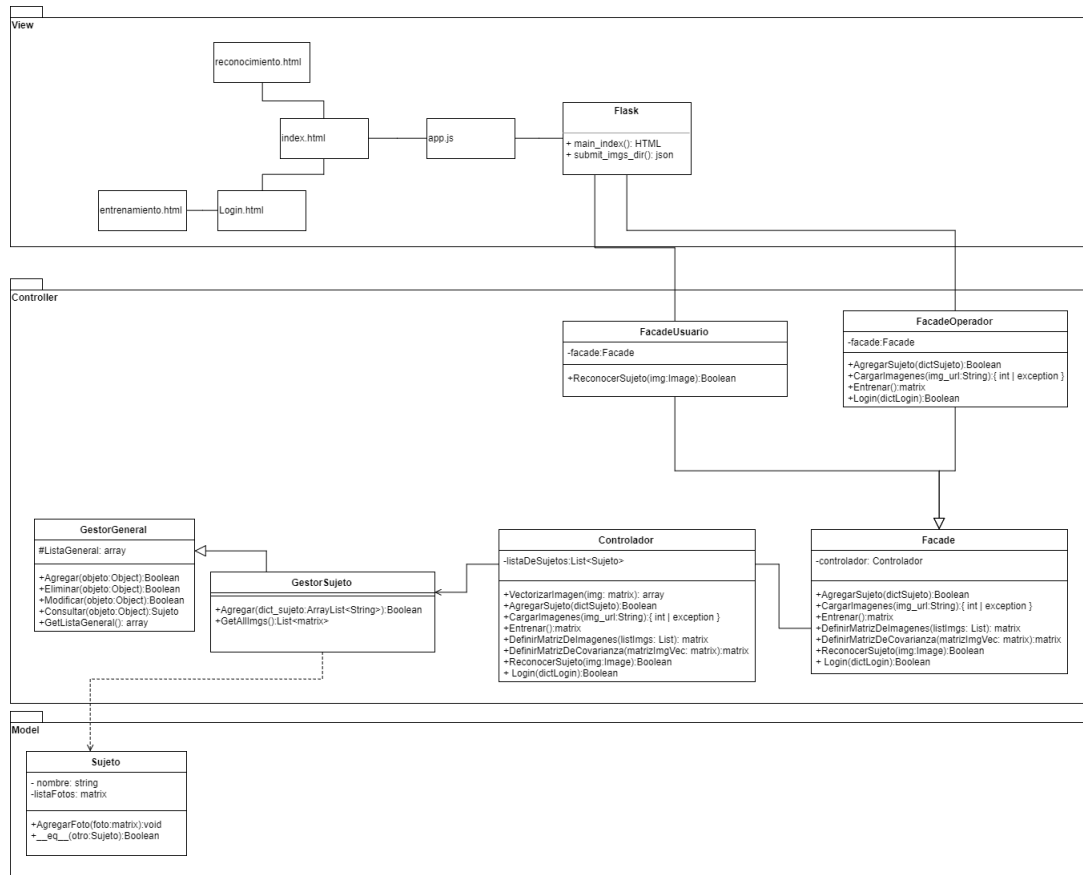


3 Diagramas

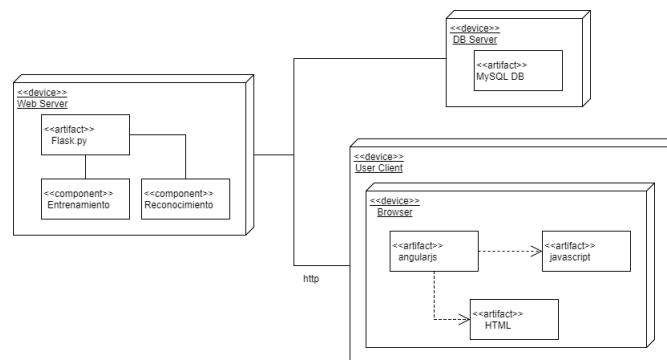
3.1 Diagrama de componentes



3.2 Diagrama de clases



3.3 Diagrama de deployment



4 Validación del diseño

Para la validación del diseño del segundo avance se utilizara el diagrama con más detalle el cual en nuestro caso es el diagrama de clases.

Unidad de diseño	Requerimiento	Relación
Gestor de sujeto	Registro de nuevos usuarios al sistema	Esta unidad será la encargada de guardar los sujetos y sus respectivas fotos en la memoria local del sistema, esto para luego ser utilizado cuando se requiera realizar el entrenamiento del sistema.
Facade	Todos los requerimientos	La función de dicha unidad de diseño será reducir la complejidad del sistema por medio de la división del mismo en otros subsistemas minimizando así la comunicación y dependencias entre estos.
Facade Usuario	Reconocimiento de un usuario	Para el caso de esta unidad se utilizará este requerimiento para separar lo que es el módulo de reconocimiento para los usuarios comunes.
Facade Operador	Registro de usuarios operadores	Para el caso de esta unidad se utilizaran estos requerimientos para separar lo que son las funciones del operador.
	Entrenamiento de varios sujetos a la vez	
	Login de usuario operadores	
	Registro de nuevos usuarios	
Sujeto	Registro de nuevos usuarios	Esta unidad de diseño será la clase encargada de representar cada sujeto una vez esté en el sistema.
Controlador	Todos los requerimientos	En este sistema se utiliza el modelo de arquitectura MVC (Modelo Vista Controlador), el cual es responsable de gran parte de la comunicación principal entre las diferentes unidades del sistema. Separando la parte de la interfaz y la lógica.

5 Especificación pruebas unitarias

5.1 test_cargar_imagenes

Entradas:

- Un string con la dirección donde se encuentre la carpeta con los sujetos y sus respectivas imágenes.
- Opcionalmente se puede agregar un parámetro (`_numParaEntrenar`) donde se especifique cuantas imágenes de cada sujeto se usarán para el entrenamiento.

Salidas:

- Una tupla con un número que representa el estado de la función y un mensaje del sistema, siendo 0 el estado de que todo ha salido bien y -1 que ha ocurrido un error.
- La lista de sujetos y sus respectivas imágenes quedan guardados en memoria para luego poder usarlos.

5.2 test_vectorizar_imagen

Entradas:

- Una matriz de números representando una imagen a vectorizar.

Salidas:

- La imagen vectorizada.

5.3 test_matriz_de_imagenes

Entradas:

- Una lista de imágenes no vectorizadas.

Salidas:

- Una matriz cuyas columnas representan cada imagen vectorizada, además a la matriz se le resta la esperanza a todos las imágenes.
- Además retorna una lista que representa la esperanza de dicha matriz.

5.4 test_matriz_de_covarianza

Entradas:

- La matriz de imagenes vectorizadas.

Salidas:

- La matriz de covarianza, formada por la multiplicación de la matriz transpuesta de imagenes vectorizadas con la matriz normal.

5.5 test_auto_valores_vectores

Entradas:

- La matriz de covarianza.
- La matriz de imagenes vectorizadas.

Salidas:

- Los auto valores y auto vectores de la matriz de covarianza ya normalizados.

5.6 test_pesos

Entradas:

- La matriz de imagenes vectorizadas.
- Los auto vectores.

Salidas:

- Una lista con los pesos de los auto vectores.

5.7 test_clasificar

Para esta prueba antes se deberá de hacer el entrenamiento pasándole la dirección de las imágenes que se desean entrenar a la función de CargarImagenes y luego llamando a la función de Entrenar.

Entradas:

- La dirección de la imagen a identificar.
- La esperanza de la matriz de imágenes.
- Los auto vectores.
- Los pesos.

Salidas:

- El sujeto que se clasificó con dicha imagen.

5.8 test_carga_de_entrenamiento

Para esta prueba antes se deberá de hacer el entrenamiento pasándole la dirección de las imágenes que se desean entrenar a la función de cargar_imagenes y luego llamando a la función de Entrenar con el prefijo "prueba_unitaria2".

Entradas:

- El prefijo del entrenamiento a utilizar

Salidas:

- Los mismos auto vectores e imagenes proyectadas que se hicieron antes de guardar el entrenamiento

Table 1: My caption

Número de Item	Tipo de Item	Detalle de Item	Nombre de Item	Nombre y versión de herramienta editora	Volatilidad
1	Diseño	Diagrama de Componentes	Diagrama de Componentes	draw.io, herramienta de edición conjunta en línea	Baja
2	Diseño	Diagrama de Clases	Diagrama de Clases ACS	draw.io, herramienta de edición conjunta en línea	Baja
3	Diseño	Diagrama de Despliegue	Diagrama de Deployment	draw.io, herramienta de edición conjunta en línea	Baja
4	Implementación	Código Fuente	Controlador.py	Eclipse 4.7 (Jee Oxygen)	Alta
5	Implementación	Código Fuente	Controlador PruebaUnitaria.py	Eclipse 4.7 (Jee Oxygen)	Alta
6	Implementación	Código Fuente	Facade.py	Eclipse 4.7 (Jee Oxygen)	Media
7	Implementación	Código Fuente	GestorGeneral.oy	Eclipse 4.7 (Jee Oxygen)	Media
8	Implementación	Código Fuente	GestorSujeto.py	Eclipse 4.7 (Jee Oxygen)	Media
9	Implementación	Código Fuente	Sujeto.py	Eclipse 4.7 (Jee Oxygen)	Media
10	Implementación	Código Fuente	Server.py	Eclipse 4.7 (Jee Oxygen)	Media
11	Implementación	Verificación de Código	Pylint		Media
12	Implementación	Servidor	Flask 0.12.2		Baja

6 Manual de Administración de la Configuración del Software

Primero se detallan los ítemes identificados:

A continuación se detalla el proceso para adquirir la configuración actual del sistema.

Los pasos siguientes son válidos para la familia de sistemas operativos Windows y versiones de Python 2.7 en adelante.

6.1 Instalación de Pip

Pip es una herramienta para la instalación de paquetes relacionados a Python. Esta herramienta es necesaria tanto para la instalación de Pylint como el servidor Flask.

El primer paso es acceder a la dirección <http://pip.readthedocs.io/en/latest/installing/#install-pip> y descargar el archivo get-pip.py.

Una vez descargado, abrir la consola de Windows y navegar a la carpeta dónde fue descargado el archivo. Siguiendo, ingresar el comando `py get-pip.py`, lo que iniciará la descarga e instalación de la herramienta Pip:

```
C:\Users\DrkSprtn\Downloads>py get-pip.py
Collecting pip
  Using cached pip-9.0.1-py2.py3-none-any.whl
Collecting wheel
  Using cached wheel-0.30.0-py2.py3-none-any.whl
Installing collected packages: pip, wheel
Successfully installed pip-9.0.1 wheel-0.30.0
C:\Users\DrkSprtn\Downloads>
```

6.2 Instalación de Flask

Una vez adquirida la herramienta Pip, se procede a instalar el servidor Flask. En la misma consola, ingresar el comando `pip install Flask`, lo que inicia la descarga e instalación del servidor Flask:

```
C:\Users\DrkSprtn\Downloads>pip install Flask
Collecting Flask
  Using cached Flask-0.12.2-py2.py3-none-any.whl
Requirement already satisfied: click>=2.0 in c:\python27\lib\site-packages (from Flask)
Requirement already satisfied: Werkzeug>=0.7 in c:\python27\lib\site-packages (from Flask)
Requirement already satisfied: Jinja2>=2.4 in c:\python27\lib\site-packages (from Flask)
Requirement already satisfied: itsdangerous>=0.21 in c:\python27\lib\site-packages (from Flask)
Requirement already satisfied: MarkupSafe>=0.23 in c:\python27\lib\site-packages (from Jinja2>=2.4->Flask)
Installing collected packages: Flask
Successfully installed Flask-0.12.2
```

6.3 Instalación de Github Desktop y Clonación del Repositorio

Una vez instalados estos componentes, se puede proceder con la instalación del cliente de escritorio Github, el cual se encuentra disponible en la dirección <https://desktop.github.com/>.

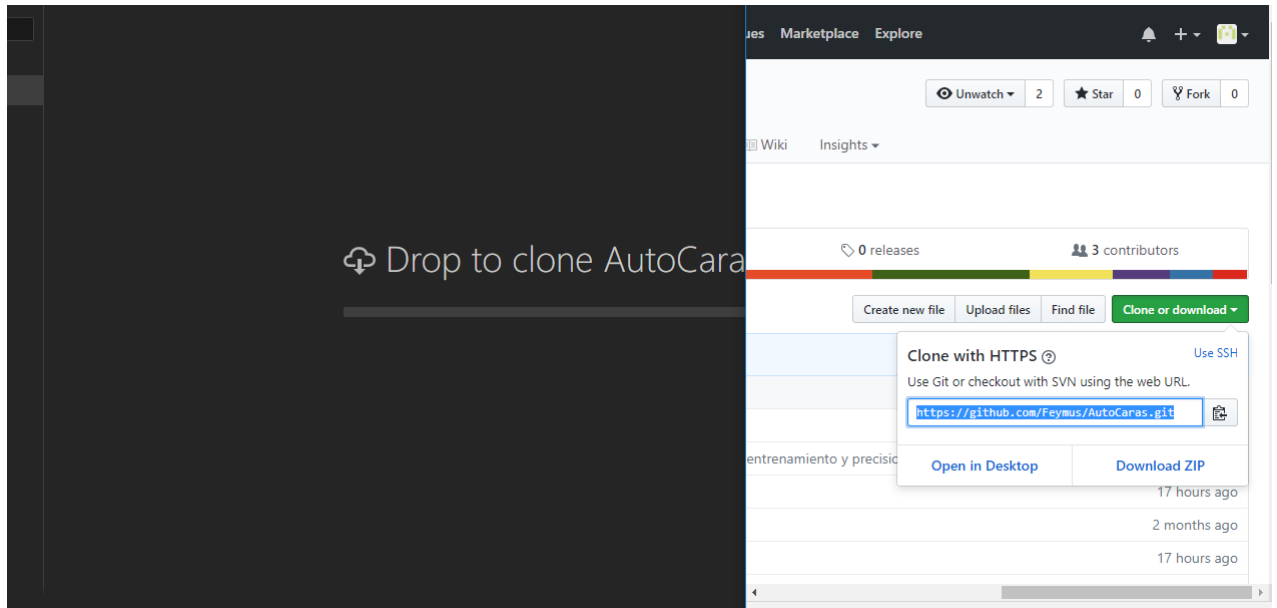
Para utilizar esta herramienta se debe poseer una cuenta en la plataforma relevante, en este caso Github.

Con esta configuración terminada, se procede a la clonación del repositorio. Primero se debe adquirir la dirección HTTPS del repositorio:

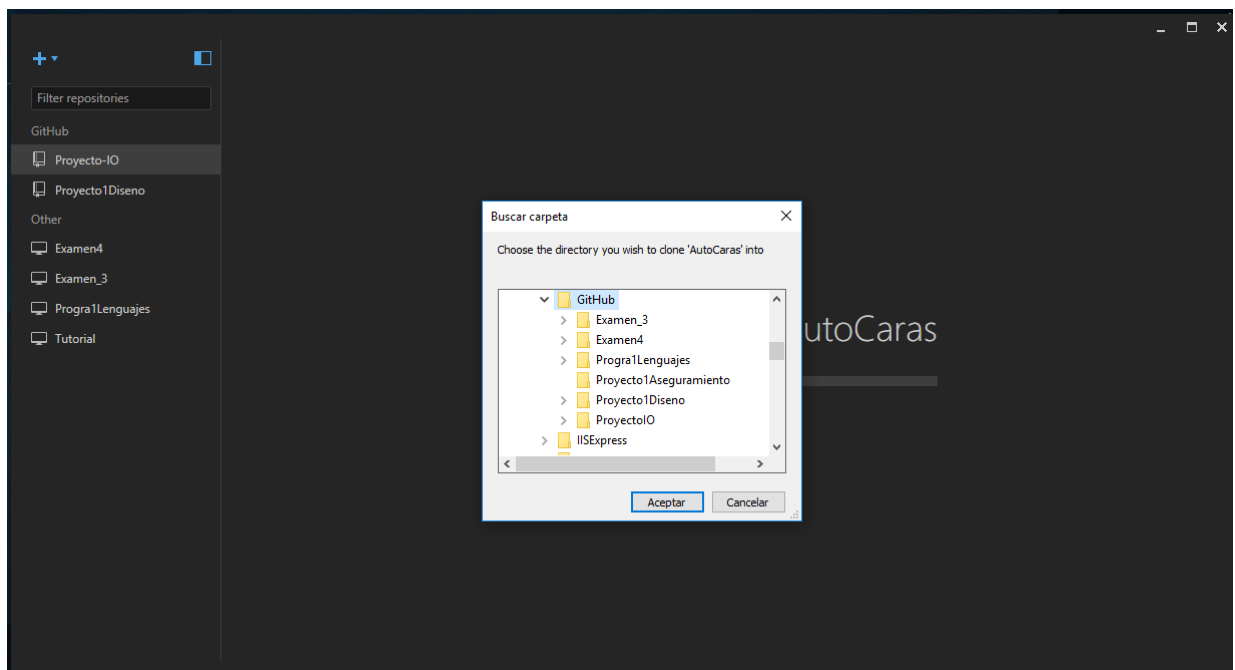
The screenshot shows the GitHub interface for the repository 'Feymus / AutoCaras'. At the top, there are statistics: 42 commits, 1 branch, 0 releases, and 3 contributors. Below this, there are tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', and 'Insights'. A message states 'No description, website, or topics provided.' Below the repository information, there is a table of files and folders. A modal is open over the 'Clone or download' button, showing the 'Clone with HTTPS' option with the URL 'https://github.com/Feymus/AutoCaras.git'. The modal also includes options to 'Open in Desktop' and 'Download ZIP'.

File/Folder	Description	Last Commit
Controller	Comentarios extra	17 hours ago
Images	Agrega nueva imagen	2 months ago
Model	Agregado interfaz para entrenamiento y precisión	17 hours ago
View	Agregado precisión	2 months ago
__pycache__	Algunas pruebas	17 hours ago
datos	Agregado precisión	2 months ago
html	Archivos doxygen y ultimo unittest	2 months ago

Para clonar el repositorio, se selecciona y arrastra la dirección hacia el cliente de github:



Al iniciar la clonación, se solicita la carpeta dónde el repositorio será clonado:



Una vez seleccionada la carpeta, el cliente iniciará el proceso de clonación. Al terminar esta descarga se cuenta con la configuración del sistema más actualizada:

.git	10/9/2017 5:32 PM	Carpeta de archivos	
__pycache__	10/9/2017 5:26 PM	Carpeta de archivos	
Controller	10/9/2017 5:26 PM	Carpeta de archivos	
datos	10/9/2017 5:26 PM	Carpeta de archivos	
Diagramas	10/9/2017 5:31 PM	Carpeta de archivos	
html	10/9/2017 5:26 PM	Carpeta de archivos	
Images	10/9/2017 5:26 PM	Carpeta de archivos	
latex	10/9/2017 5:26 PM	Carpeta de archivos	
Model	10/9/2017 5:26 PM	Carpeta de archivos	
View	10/9/2017 5:26 PM	Carpeta de archivos	
.project	10/9/2017 5:26 PM	Archivo PROJECT	1 KB
.pydevproject	10/9/2017 5:26 PM	Archivo PYDEVPR...	1 KB
ImageRecognition-AC.doxyfile	10/9/2017 5:26 PM	Archivo DOXYFILE	106 KB
README.md	10/9/2017 5:26 PM	Archivo MD	1 KB
tiempo.py	10/9/2017 5:26 PM	Python File	1 KB
Trozos.txt	10/9/2017 5:26 PM	Documento de tex...	1 KB

7 Actividades

Actividad	Responsable	Etapa
Verificar la coherencia entre las necesidades del cliente y los requerimientos solicitados	Nelson Gómez	Análisis
Comparación del diseño contra los requerimientos funcionales y no funcionales	Michael Choque Núñez	Diseño
Verificación de wireframes con el cliente	Nelson Gómez	Diseño
Verificación de prototipos con el cliente	Nelson Gómez	Diseño
Comparación de la implementación contra el diseño	Michael Choque Núñez	Implementación
Comparación entre los requerimientos y la implementación	William Espinoza Ramírez	Implementación
Evaluación de la conformidad del código Python respecto al estándar PEP-8	William Espinoza Ramírez	Implementación
Evaluación de la conformidad del código javascript respecto al estándar Google Style	William Espinoza Ramírez	Implementación
Desarrollo y aplicación de pruebas unitarias	Michael Choque Núñez	Pruebas
Desarrollo y aplicación de pruebas de integración	Michael Choque Núñez	Pruebas
Pruebas de conformidad con el cliente	William Espinoza Ramírez	Pruebas
Verificar la conformidad con los umbrales de precisión y exactitud	Nelson Gómez	Pruebas

8 Métricas implementadas

8.1 Eficiencia de carga de los archivos

El requerimiento asociado a dicha métrica es al del reconocimiento preciso de un usuario. La cuál estipula que el sistema debe dar una respuesta al usuario de manera precisa y eficiente. La herramienta que lo mide es una función llamada `measure time`.

Métrica de eficiencia - Measure Time

```
import time

def measure_time(f):
    def timed(*args, **kw):
        ts = time.time()
        result = f(*args, **kw)
        te = time.time()
        print ('%r (%r, %r) %2.2f sec' % \
              (f.__name__, args, kw, te-ts))
        return result
    return timed
```

Cómo utilizar la herramienta

Primero, la función debe incluirse en el archivo a probar, mediante un import. Siguiente, se debe escribir la etiqueta @tiempo.measure_time antes de la función cuyo tiempo se desea medir. Al ejecutarse el código, se imprimirá en consola el tiempo consumido por la función especificada:

```
Server.py [C:\Python27\python.exe]
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
'cargar_imagenes' ((<Controller.Controlador.Controlador object at 0x04561470>, u'C:\\Users\\DrkSprtn\\Documents\\GitHub\\AutoCaras\\Images'), {}) 0.33 sec
```

Resultados Obtenidos

Table 2: Pruebas de eficiencia en la carga de archivos

# Corrida	Tiempo	# Corrida	Tiempo
1	0.06s	11	4.65
2	0.06s	12	5.65
3	0.06s	13	0.06s
4	0.07s	14	0.06s
5	0.06s	15	0.06s
7	0.06s	16	0.06s
8	0.06s	17	0.06s
9	0.06s	18	0.06s
9	0.06s	19	0.06s
10	0.06s	20	0.06s
Promedio		0.5645 s	
Total			

8.2 Métrica de precisión

Métrica de precisión - Exactitud a la expectativa

El requerimiento asociado a dicha métrica es al del reconocimiento preciso de un usuario. Se utilizará una formula como herramienta manual para obtener la exactitud a la expectativa.

Cómo utilizar la métrica

Se utilizará la siguiente formula:

$$x = A / T$$

Donde:

A = Número de casos encontrados por los usuarios con una diferencia a lo esperado.

T = Número de operaciones realizadas.

Entre X más se acerque a 0, más preciso es el sistema. $0 \leq X$. En este caso se hacen varias pruebas de reconocimiento de sujetos donde por ejemplo:

SR, s1 , S, s1

En el caso anterior, es un caso exitoso del sistema, donde s1 es el sujeto real que fue está siendo verificado y el segundo s1 es el resultado del sistema.

SR, s7 , S, s10

Mientras que en este otro caso, el s7 fue el sujeto real y s10 el resultado determinado por el sistema, lo cuál es un error y se suma a la A utilizada por la formula.

Resultados Obtenidos

Se hicieron 328 pruebas en total.Y se encontraron 84 número de casos diferentes al esperado. Así:

$$A = 84 \quad T = 328$$

$$A / T = X \rightarrow 84 / 328 = 21/82 = 0.256097$$

Por lo tanto X tendrá un valor de 0.256097 que es la exactitud a la expectativa.