

Instituto Tecnológico de Costa Rica

Ingeniería en Computación

IC1803 - Taller programación

Tarea 1: Atributos de calidad y métricas

Profesor: Saúl Calderón Ramírez

Estudiante:

José Miguel Mora Rodríguez

Dylan Rodríguez Barboza

Karina Zeledón Pinell

Segundo semestre 2017

Contents

1. Requerimientos del Sistema	3
2. Contexto del proyecto	3
3. Categorías y atributos elegidos para cada una	3
3.1. Atributos y su prioridad	4
3.1.1. Funcionalidad	4
3.1.2. Fiabilidad	4
3.1.3. Eficiencia	4
3.1.4. Usabilidad	5
3.1.5. Portabilidad	5
3.1.6. Mantenibilidad	5
3.2. Atributos y sus métricas	6

1. Requerimientos del Sistema

Para identificar los atributos de calidad pertenecientes a cada categoría que se requieren para este proyecto, primero se deben identificar los requerimientos más importantes:

1. Brindar un medio en el que se puedan almacenar las muestras de cada persona.
2. Brindar un medio en el que se puedan almacenar los datos personales de cada persona.
3. Crear un sistema capaz de ejecutarse en los equipos de diferentes entradas del país (aeropuerto Juan Santamaría, Peñas Blancas, Paso Canoas, Sixaola).
4. Crear un sistema capaz de encontrar similitudes entre el sujeto que se está identificando y los datos almacenados.
5. Capacidad de informar al usuario cuando no se encuentre una coincidencia.
6. Capacidad de usar los datos no coincidentes del punto 5 para así poder entrenar al sistema.
7. Capacidad de almacenar los datos de huellas dactilares.
8. Capacidad de almacenar los datos del iris ocular.
9. Capacidad de buscar coincidencias de los datos en los puntos 7 y 8 en caso de que el reconocimiento facial falle.
10. Capacidad del sistema de funcionar en diferentes entornos.

2. Contexto del proyecto

Se requiere de un sistema de reconocimiento facial que pueda interactuar con otros dos complementarios (reconocedor de huellas dactilares y de iris ocular). Es importante que el sistema funcione en diferentes entornos, por lo que el lenguaje de programación elegido para este fin es python, el cual es conocido por ser multiplataforma. El IDE seleccionado para la creación de scripts en este lenguaje es eclipse, bajo el entorno de PyDev.

Se requiere que el sistema tenga alta disponibilidad para usuarios de diferentes entradas del país, por lo que es conveniente la creación de una aplicación web. Para este fin, se seleccionó un framework llamado django que actúa en conjunto con el lenguaje de programación python, el aseguramiento de disponibilidad de esta implementación del sistema se puede ver en los atributos de calidad definidos a lo largo del documento.

3. Categorías y atributos elegidos para cada una

Cabe destacar que para el contexto del proyecto (realización de una versión beta), los umbrales para las métricas están pensados para un número pequeño de usuarios, por ejemplo, el número de estudiantes pertenecientes al Instituto Tecnológico de Costa Rica.

3.1. Atributos y su prioridad

3.1.1. Funcionalidad

Subcaracterística	Descripción	Peso (Alto/Medio/Bajo)
Exactitud	El sistema requiere de exactitud en sus resultados, con tal de brindarle al usuario la aproximación más correcta al rostro que se quiere reconocer	A
Interoperabilidad	El sistema requiere hacer interacción con otros sistemas de reconocimiento para complementar el reconocimiento de rostros	M
Seguridad de acceso	El sistema requiere que solo los usuarios autorizados puedan acceder a las funcionalidades principales de este	A

3.1.2. Fiabilidad

Subcaracterística	Descripción	Peso (Alto/Medio/Bajo)
Madurez	El sistema requiere que los fallos que más persistan sean tratados, con su posible eliminación, con tal de brindar el mejor servicio y aproximación que requiere el usuario	A
Tolerancia a fallos	El usuario requiere que el sistema pueda realizar operaciones o que al menos pueda notificar cuando está sucediendo un fallo, de tal forma que este pueda tomar alguna medida para solucionar el problema	A
Capacidad de recuperación	El usuario requiere que el sistema sea capaz de brindar de nuevo el servicio lo más rápido posible después de una falla en este.	M

3.1.3. Eficiencia

Subcaracterística	Descripción	Peso (Alto/Medio/Bajo)
Comportamiento temporal	El sistema debe dar las aproximaciones de reconocimiento en el menor lapso posible, la rapidez de respuesta es esencial	A
Utilización de recursos	El sistema debe utilizar los recursos de manera que no los desperdicie, más específicamente el espacio en disco que requiere el guardado de los datos con los que se entrena al sistema	M
Cumplimiento de eficiencia	El sistema requiere cumplir con un estándar de eficiencia que puede ser medido mediante pruebas y documentaciones de las mismas	M

3.1.4. Usabilidad

Subcaracterística	Descripción	Peso (Alto/Medio/Bajo)
Capacidad para ser aprendido	El sistema debe ser fácil de aprender para el usuario, de manera que, en el segundo uso de la aplicación, el usuario ya pueda recordar cómo funcionaba	A
Capacidad para ser entendido	Las funcionalidades del sistema (pedido de datos, solicitudes) deben ser intuitivas en la interfaz gráfica de usuario, de manera que el mismo pueda entender lo que está sucediendo	A
Capacidad para ser operado	El sistema debe ser de fácil uso para el usuario, de manera que este pueda hacer operaciones brindadas por el sistema de manera rápida y exitosa	A

3.1.5. Portabilidad

Subcaracterística	Descripción	Peso (Alto/Medio/Bajo)
Adaptabilidad	El sistema debe ser capaz de ejecutarse en varios entornos, es decir, tener la capacidad de adaptarse a diferentes ambientes (multiplataforma).	A
Instabilidad	El sistema debe ser instalado con facilidad en cualquier ambiente, al igual que el atributo de calidad anterior, debe ser multiplataforma	A
Capacidad para ser reemplazado	El sistema debe contener módulos bien estructurados, de tal manera de que la sustitución de uno de estos por otro que haga tareas parecidas no sea un impedimento para la adecuada ejecución y generación de resultados	M

3.1.6. Mantenibilidad

Subcaracterística	Descripción	Peso(Alto/Medio/Bajo)
Capacidad para ser analizado	El sistema debe ser abierto a diversos análisis que puedan determinar si hay una inconsistencia en este, de manera que se pueda planear una solución sencilla a esta	M
Capacidad para ser cambiado	El sistema debe contener módulos que eviten tener en la medida de lo posible errores de software o inconsistencias, de manera que cada módulo pueda ser cambiado en las diferentes modificaciones que se le haga	A
Estabilidad	El sistema debe seguir funcionando correctamente aún cuando se hayan cambiado ciertos módulos al darle mantenimiento	A

3.2. Atributos y sus métricas

Característica	Métrica	Descripción	Herramienta	Nivel Requerido	Evaluación real de los resultados
Exactitud	Exactitud a la expectativa	Realización de pruebas de unidad en las que se compruebe que muchas pruebas podrán generar los resultados que se esperan	Selenium	Que el 90 % de los resultados reales coincidan con los resultados esperados, mediante dos o tres pruebas por funcionalidad	
Interoperabilidad	Intercambio de datos (Formato basado en datos)	Realización de diagramas que muestren la funcionalidad del sistema de reconocimiento de caras en conjunto de otros como lo es el de reconocimiento de huellas dactilares o del iris ocular	Visio	Un diagrama de componentes por cada sistema que se implemente. Un componente por cada módulo que se quiera implementar	
Seguridad de acceso	Auditabilidad de acceso	Hacer una medición de seguridad periódica para conocer el rumbo que toma la seguridad en el código con el que se está implementando el sistema, más específicamente un inicio de sesión al sistema	Fortify	Que el 90 % de los módulos del sistema hayan tenido dos o tres pruebas de seguridad	
Madurez	Detención de fallas	Creación de un documento que resuma las fallas que persisten en el sistema (mediante un diagrama que ejemplifique los componentes), con tal de que sirvan como documentación de referencia para que pueda ser resuelto con la mayor brevedad	Visio	Que sea el 10 % el máximo de módulos que presenten alguna falla (representados mediante componentes en los respectivos diagramas)	

Tolerancia a fallos	Prevencion de averías	Realización de pruebas que puedan hacer que el sistema experimente fallos, esto con el fin de plantear soluciones que le permitan a este actuar ante estas situaciones	Selenium	Dos o tres pruebas en las que el sistema pueda fallar. Que el 90 % de los módulos presenten la tolerancia deseada. Que el sistema pueda seguir realizando ciertas operaciones después de un fallo	
Capacidad de recuperación	Restablecimiento	Realización de pruebas periódicas en las que se pueda evaluar cuánto tiempo tarda el sistema en volver a brindar los servicios esperados después de un fallo	Selenium	Cada vez que se implemente un módulo, grabar dos o tres pruebas para el mismo, esto para asegurarse que al menos el 90 % de los módulos experimentan una buena recuperación	
Comportamiento temporal	Tiempo de Respuesta	Al ser un sistema pensado para ser puesto en producción en controles fronterizos, teniendo en cuenta el número elevado de personas que deberán utilizar el sistema, es importante que el tiempo que toma en lograr reconocer una imagen sea relativamente corto	Eclipse (Pydev)	Tiempo de respuesta entre 5 y 10 segundos	

Utilización de recursos	Máxima utilización de la memoria	Mediante pruebas realizadas en el POC se ha comprobado que el uso de disco en este tipo de sistemas es bastante elevado, por lo que el lograr la reducción del espacio utilizado en disco es de suma importancia	Visio	Documentación del uso de disco de cada nuevo módulo (representado por un componente), para asegurarse de que el 90 % de estos haga un uso adecuado del mismo	
Cumplimiento de eficiencia	Cumplimiento de la eficiencia	Realización de pruebas a las que se les pueda medir el tiempo de ejecución, para saber con mayor aproximación la eficiencia de cada módulo que se quiera probar	Selenium	Realizar dos o tres pruebas a cada nuevo módulo que se implemente. Asegurarse de que el 90 % de módulos cumplan con la eficiencia	
Capacidad para ser aprendido	Facilidad en función del aprendizaje	Realizar pruebas con un número de usuarios que puedan hacer uso del sistema, esto con el fin de evaluar la facilidad que puedan experimentar para utilizarlo	Eclipse (PyDev)	Realizar dos o tres pruebas con los usuarios cada vez que se implemente una nueva sección navegable, de manera que se pueda asegurar de que todos los usuarios puedan aprender fácilmente las funcionalidades del sistema	
Capacidad para ser entendido	Capacidad de demostración	Hacer un análisis de los escenarios de uso para evaluar secciones navegables que faciliten el uso del sistema por parte de los usuarios	Visio	Asegurarse de que el 100 % de las secciones navegables sean documentadas en un diagrama	

Capacidad para ser operado	Claridad del mensaje	Realización de un diagrama de navegación que le facilite a cualquier equipo de trabajo involucrado la comprensión y realización de una interfaz gráfica en la aplicación que sea comprensible para el usuario y que le permita realizar las tareas que brinda la aplicación fácilmente	Visio	Agregar un componente al diagrama de navegabilidad cada vez que se implemente una sección navegable, de forma que el 100 % de estas sean navegables	
Adaptabilidad	Adaptabilidad del Software	Desarrollo del sistema en un lenguaje de programación multiplataforma. El lenguaje de programación python es adecuado para este propósito	Eclipse (PyDev)	Dos o tres pruebas en diferentes entornos para asegurarse de que el 90 % de los módulos funcionen correctamente	
Instabilidad	Facilidad de instalación	Al igual que en la adaptabilidad, es esencial el desarrollo del sistema en un lenguaje de programación multiplataforma. Para este caso, python es adecuado para el manejo de imágenes, además, este brinda un entorno multiplataforma	Eclipse (PyDev)	Dos o tres pruebas en diferentes entornos para asegurarse de que el 90 % de los módulos funcionen correctamente	
Capacidad para ser reemplazado	Función de inclusividad	Documentación de los módulos del sistema que pueden ser reemplazados o ser usados en un entorno diferente, esto permitirá que otros equipos que trabajen en el sistema, logren hacer los reemplazos de manera exitosa	Visio	El 90 % de los componentes deben especificar bajo qué entornos pueden usarse	
Capacidad para ser analizado	Capacidad de análisis de fallos	Realización de períodos de análisis de código en las que se puedan encontrar errores en el mismo, especialmente detección de pulgas en el sistema	Sonar Qube	Hacer revisión del código cada vez que se implementen tres módulos en el sistema, de manera que se pueda asegurar el 90 % del total de estos	

Capacidad para ser cambiado	Complejidad de modificación	Realización de períodos de análisis de código en las que se puedan encontrar inconsistencias en el mismo, tales como los olores de software	Sonar Qube	Hacer revisión del código cada vez que se implementen tres módulos en el sistema, de manera que se pueda asegurar el 90 % del total de estos	
Estabilidad	Tasa de cambio exitoso	Realizar pruebas periódicas de las funcionalidades del sistema cada vez que se cambien cierta cantidad de módulos, con el fin de asegurarse de que el sistema siga funcionando como se espera	Unittest (módulo de python)	Hacer pruebas de unidad cada vez que se implemente o cambie un módulo, de manera que se asegure el funcionamiento del 90 % de estos	