

Instituto Tecnológico de Costa Rica

Ingeniería en Computación

IC6831 - Aseguramiento de Calidad

Proyecto semestral: Prueba de concepto

Profesor: Saúl Calderón Ramírez

Estudiantes:

Luis Alonzo Cascante Franco 2014159896

Olman Castillo Picado 2015148651

María Laura Pizarro Moreno 2015120626

Gabriel Venegas Castro 2013115967

Segundo semestre 2017

1. Introducción

Una prueba de concepto es una implementación mínima de una idea o concepto. Esta etapa será utilizada para probar la factibilidad de las ideas, tecnologías y algoritmos para resolver las necesidades básicas del proyecto semestral. Se desarrollarán 6 tozos; cargar imágenes, pasar de color a blanco y negro las imágenes utilizadas por el programa, vectorizar imágenes, crear matriz con los vectores creados, cambiar las dimensiones de una imagen y calcular matrices de covarianza. Además se utilizará la herramienta Zoho Projects para implementar el método Scrum y Github para manejar el control de versiones. Se van a hacer pruebas unitarias de cada trozo en el programa Eclipse utilizando el lenguaje Python para el back-end y Django para hacer el front-end.

2. Análisis del problema

El problema propuesto será descompuesto en sub-problemas, además de tomar en cuenta los requerimientos funcionales especificados en tal descomposición del problema.

1. Cargar imágenes: El usuario va a poder cargar una imagen desde el front-end.
2. Pasar de color a blanco y negro las imágenes: Toda imagen que el programa manipule, ya sea de la base de datos o ingresada por el usuario se va a convertir a una escala de grises.
3. Vectorizar imágenes: Todas las imágenes que el programa utilice van a ser vectorizadas, lo que significa que en cada vector se va a almacenar el valor de cada pixel de cada imagen.
4. Crear matriz con las imágenes vectorizadas: Luego de vectorizar las imágenes dichos vectores se van a almacenar en una misma matriz.
5. Cambiar las dimensiones de una imagen: Toda imagen que el usuario ingrese al sistema, sus dimensiones van a ser modificadas para que no haya conflictos en el programa.
6. Cálculo de matrices de covarianza: Es necesario calcular la matriz de covarianza para determinar cuáles de los píxeles presentes en las imágenes son los más relevantes para el algoritmo de reconocimiento de caras. La matriz de covarianza permite saber qué dimensiones aportan la menor cantidad de información para ser desechadas al llevar a cabo la reducción de dimensiones.

3. Diagrama de componentes

El programa se compone de los siguientes componentes:

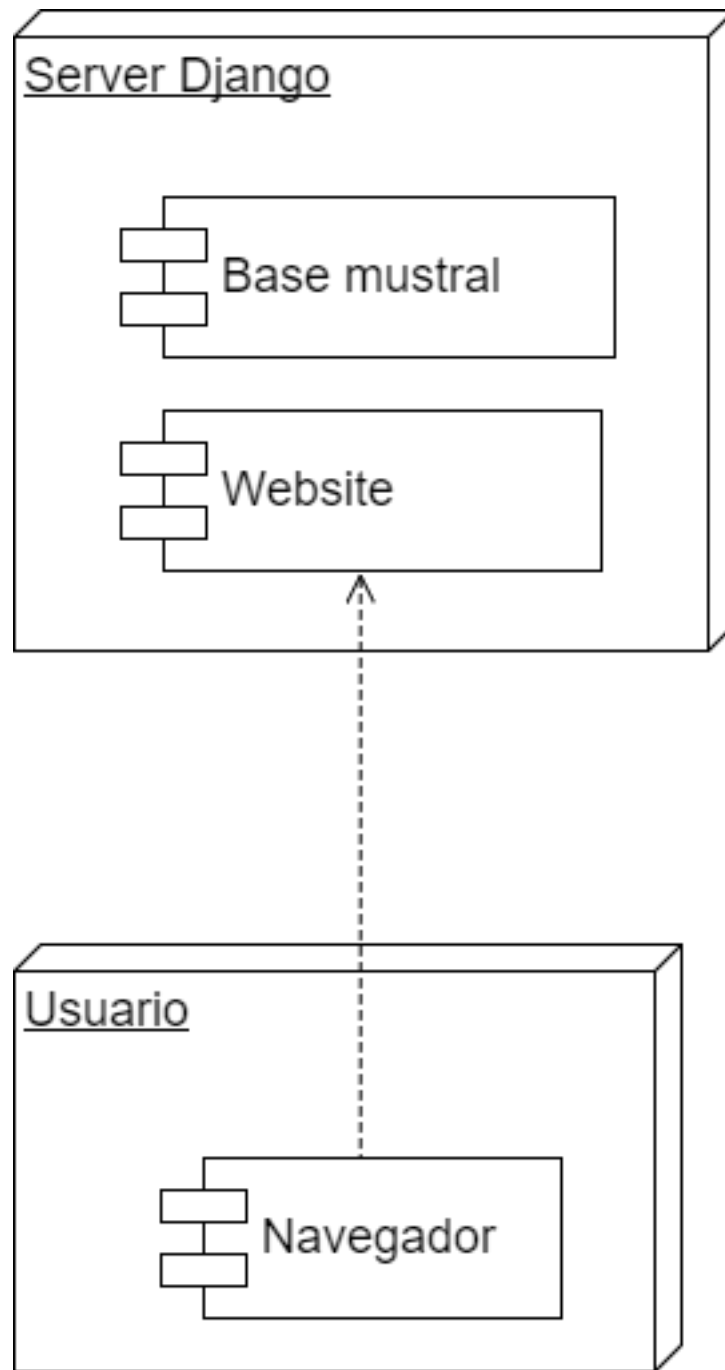


Figura 1: Diagrama de componentes

4. Implementación y pruebas

El documento que especifica las pruebas, los métodos, variables y demás se encuentra por aparte ya que se utilizó la herramienta de doxygen que trabaja con javadocs para crear archivos LaTeX/HTML que contienen la información del proyecto.

5. Conclusiones y recomendaciones

Todos los trozos tuvieron resultados satisfactorios ya que se implementaron correctamente y sin ningún problema que impidiera su correcto funcionamiento. Se utilizó la librería OpenCV para la manipulación de las imágenes y numpy para los cálculos matemáticos complejos. Al programa le faltan más requerimientos por ser implementados, los cuales se realizarán en las siguientes entregas por realizar. Durante el desarrollo de los trozos se presentaron algunos problemas los cuales se mencionan a continuación:

Windows 10 y Django:

- Problema: Se descubrió que en Windows 10 surgen errores cuando se quiere trabajar desde el cmd ya que no reconoce Django con el error de que Django no está instalado a pesar de que sí lo esté.
- Resolución: Luego de tratar de arreglar el error para seguir trabajando desde el cmd se logró encontrar otra alternativa que Eclipse brindaba. Eclipse tiene la opción de ejecutar los mismos comandos que se usarían en el cmd solo que desde Eclipse no se generaba ningún problema.

CSS y JS:

- Problema: Al inicio se desconocía que Django no leía los documentos .html como se leen usualmente en el desarrollo web.
- Resolución: Se hizo una investigación para averiguar la sintaxis correcta para incluir los archivos css y js.

Lyx:

- Problema: Cuando se abrió el programa por segunda vez empezó a mostrar mensajes de que habían módulos que no se encontraban y cada cierto tiempo volvía a mostrar el mismo error, además a algunos integrantes del grupo les presento problemas abrir el documento en el que se estaba trabajando, ya que LyX indicaba de que no se pudieron encontrar ciertos archivos necesarios para la ejecución. Dicho problema imposibilitó que todos los integrantes pudieran trabajar en el documento.
- Resolución: Se decidió que los que no podían abrir el documento del todo pasaran la información por aparte y que el que sí podía abrir el documento agregara dicha información. El error del mensaje constante de los módulos no encontrados no se pudo solucionar pero dicho problema no impidió que se trabajara en el documento.

OpenCV

- Problema: A pesar de que todos los integrantes del grupo instalaron la misma versión de OpenCV (3.1.0), no a todos les funcionaba correctamente, provocando que no todos pudieran probar el programa completo
- Resolución: A los que sí les funcionaba OpenCV correctamente les tocó asegurarse de que lo que los otros integrantes programaban sirviera correctamente.