

Aseguramiento de la Calidad del Software

Documentación POC

Javier Contreras Muñoz, Randall Delgado Miranda, Sebastián Porras Fallas

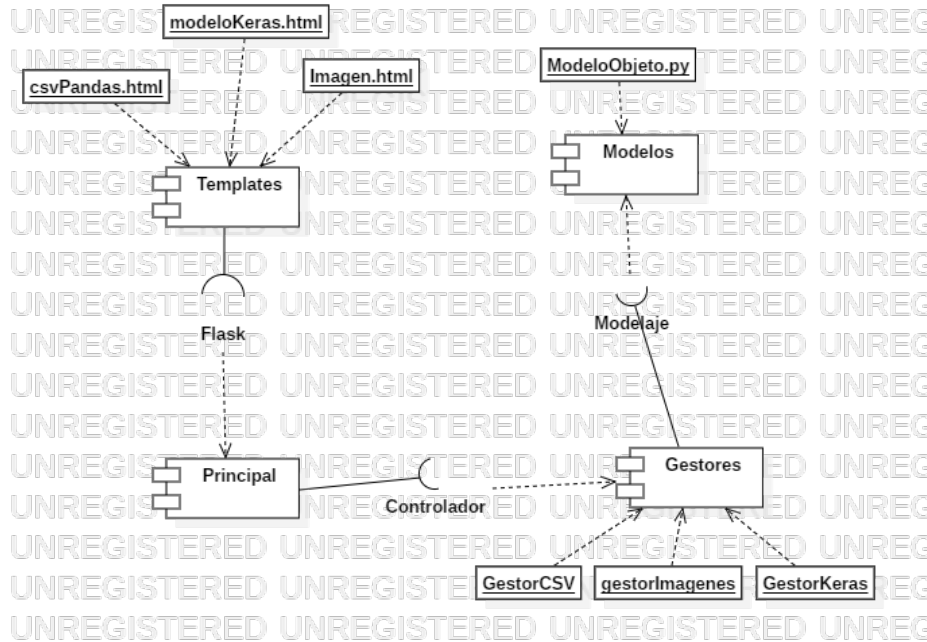
August 23,2018

Abstract

Proof of Concept is a minimum implementation of an idea or concept. The usual objective of this one is to evaluate and test the feasibility of an idea implementing several technologies/algorithms with the objective of solving an specific problem or necessity

The following document will include a Component Diagram related with the Proof of Concept project. Also a brief analysis of each of the functionalities will be included. Finally, it will be shown a description of some of the encountered problems in the used technologies including the solution for it

1 Diagrama de Componentes



2 Análisis de los Trozos Implementados

- **Generación de CSV:** En esta prueba de concepto, la página web, siguiendo un poco el contexto de la aplicación, provee al usuario la capacidad de poder generar un CSV el cual contiene los datos de objetos de una imagen. Entre estos datos cabe destacar que se tienen un identificador, un área y un centroide.

Analizando desde el punto de vista del código, el usuario puede ingresar la información de hasta n objetos, convirtiéndose estos en una lista de diccionarios. Dicha lista va a ser transformada con el objetivo de extraer en listas por separadas, cada uno de los atributos de dichos objetos. Una vez realizado esto, se procede a crear un nuevo diccionario con cada una de las listas creadas con el fin de utilizar la librería de Pandas para escribir un CSV con la información a un directorio de salida específico.

Cabe destacar que el uso de la librería Pandas facilita en gran medida el manejo de los CSV ya que se requieren muy pocas líneas de código para realizar las distintas acciones, por ejemplo, la escritura de varios datos. Asimismo, esto aumenta, en cierta parte, la legibilidad del código.

- **Cargar y Guardar Imágenes:** La página web ofrece un módulo para la carga de imágenes. Para cargar la imagen, esta necesita estar en el directorio "static" del proyecto. La razón de esto es que el sistema de archivos no provee el path completo a la hora de seleccionar un archivo, solo el nombre, por lo que se debe escoger una carpeta específica. En este caso va a ser static debido a que ahí se almacenan todos los recursos y demás de un proyecto de tipo Flask.

A nivel de código lo que sucede es que se agarra el nombre del archivo y en el backend se verifica que el nombre es correcto, es decir, que existe como tal en la carpeta static. De ser así, utilizando la librería PIL se abre el archivo y se procede a hacerle ciertas transformaciones con la librería numpy con el objetivo de poder guardarla. En el directorio de salida que el usuario haya digitado en la aplicación.

Cabe destacar que el uso de numpy y PIL para manejo de imágenes no requiere de un análisis profundo de las librerías como tal.

- **Implementación de Keras:** Keras es una librería que permite el entrenamiento de modelos para reconocer patrones, y es muy utilizado especialmente en imágenes. Al incluir integración con Python, se vuelve sumamente útil para trabajar en conjunto con otras librerías como numpy y matplotlib.

Para la presente prueba de concepto, se ejemplificó su uso mediante la carga y guardado de un modelo definido por el usuario. En esta funcionalidad se permite al usuario definir parámetros como el número de filtros, los modos de activación de las capas (layers), las especificaciones para compilarlo, el número de épocas y el tamaño de los bloques para

trabajar.

Por dentro, la página Web envía los datos recopilados a una función que se encarga de tomar los valores, armar un modelo, y crear un archivo JSON a partir de éste, así como un archivo H5 donde se guardan valores de pesos de prueba. Para cargarlo, se busca tal modelo por su nombre, y se cargan ambos archivos a memoria.

3 Problemas Encontrados con las Herramientas

- **Flask:** El primer problema encontrado fue que nunca se había utilizado Flask entonces se tuvo que sacar el tiempo para aprender a utilizarlo. Con la ayuda de tutoriales de la página oficial se logró cumplir el objetivo.

Por otro lado, se descubrió que Flask utiliza los "HREF" de manera distinta. Si bien en otros frameworks se permite simplemente poner el directorio en donde se encuentra el HTML, en Flask se debe redireccionar a un `app.route()` específico. En un principio, esto se tornó en un problema debido a que no se sabía que era lo que estaba sucediendo y de cierta forma atrasó el desarrollo del proyecto.

La solución utilizada fue la de realizar una llamada a un form y dependiendo de la selección del usuario, se va a desplegar el HTML correspondiente.

- **Eclipse:** Se encontraron distintos problemas con el plugin de git para Eclipse, el principal es que la opción de manejo de conflictos no es tan fácil de usar, por lo que muchas veces implica tener que hacer merge de los documentos en conflicto de forma manual.
- **Herramienta de Doxygen:** Uno de los problemas encontrados con la herramienta Doxygen es que el repositorio oficial para descargarla se encuentra caído.

La solución principal ante este problema fue el uso de un repositorio de

GitHub con el fin de obtener el software, sin embargo, al utilizar repositorios no oficiales existe la posibilidad de que la versión del software no sea la más reciente o bien, sea dañina para la computadora.

- **TensorFlow:** Se encontraron problemas al momento de la instalación ya que TensorFlow está hecho para trabajar con Python 64 bits, específicamente las versiones: 3.5.x y 3.6.x

Se solucionó al instalar una versión anterior de Python (3.6.6).