El dia 11 de Junio tuvimos una nueva reunión, una ya de las ultimas en teoría para poner en común los avances e ir enfocando como terminar el trabajo.

Antes de nada como siempre indicar lo realizado antes de la reunión. Domingo no pudo solucionar los problemas que había tanto con el filechooser como con el pop up, aunque seguirá trabajando en ello. Por mi parte desarrolle el punto de la memoria de simulaciones incluyendo las graficas resultantes, y alguna conclusión que se saca de esto, con lo que este punto esta cerrado a falta de revisar la memoria. Por otro lado en los otros dos puntos, tanto en el de diseño como en el de pruebas encontré dificultades que íbamos a observar en esta reunión.

Como ya he dicho, nos centramos en la memoria en esta ocasión ya que faltaban de corregir las cosas que le faltaban al programa en si.

Y es que tras revisar distintos TFGs observaba distintos graficos que no sabia si eran de utilidad y el como añadirlos a la memoria. Estos graficos y esta parte que faltaba se centraba en los capítulos de diseño arquitectónico y de pruebas.

Por un lado en el capitulo de diseño el problema estaba en las distintas capas, las cuales ya he explicado pero que falta indicar concretamente para nuestro TFG como funciona cada una de ellas. Por ejemplo en la capa de datos hay que añadir un diagrama de base de datos con los datos de nuestra base de datos, indicando los atributos y las relaciones. En nuestro caso este diagrama será muy pequeño porque la base de datos es pequeña donde una foto puede tener distintos datos (dependiendo de los resultados) pero unos datos solo pueden corresponder a una foto. Haremos ese diagrama (Con BOUML como los casos de uso) y una captura de cómo esta la base de datos en flaskblog.py.

Importante diferenciar que la capa de datos no tiene los datos mientras se ejecutan en el programa ya que el programa solo esta vinculado a la base de datos cuando pasas los datos al servidor. Estos datos que entran al programa por parte del usuario y que a partir de ahí se utilizan para calcular la varianza estarían mas en la capa de negocio que es la que se encarga de detallar lo que hace el programa.Lo importante en esta capa es explicar los métodos, las funciones, etc con capturas del código en si. En cualquier caso revisare los tfgs por si creo que hace falta añadir o cambiar algo.

Por ultimo la capa de presentación que básicamente tiene que dejar ver lo que se hace en cada escena, las opciones que tiene el usuario incluso el diagrama de flujo que tendría la aplicación según la acción del usuario. Para ayudar con esta capa realizaremos mockups de lo que seria la interfaz en distintos momentos. Un mockup es como un boceto de la interfaz. Para hacerlo utilizaremos el programa Axure, que podemos probar durante 30 dias de forma gratuita. Suficiente para realizar estos diseños. Esta seria la parte de diseño y solo queda avanzar con ello.

Despues quedaría añadir la parte de las pruebas, recordando las diferencias entre cada prueba. Las pruebas unitarias eran las que se hacían de forma independiente para una funcionalidad, como pueden ser las pruebas de la varianza o las pruebas de la interfaz (ya descritas en la memoria). Cualquier prueba para la que hemos utilizado un programa externo como por ejemplo últimamente con el pop up, o anteriormente el generador, seria una prueba unitaria esperando un buen desarrollo de ese punto.

Las pruebas de integración seria añadir cada modulo al programa y ver que el re4sultado es el mismo. Por ejemplo con el pop up, hemos tenido que hacer cambios ya que la prueba se hizo con un pop up y la aplicación, pero a la hora de integrarlo en el programa serán dos pop ups. Lo mismo con la varianza, no es lo mismo probar la varianza con datos de ejemplo, que probarlos con unos valores de entrada que desconocemos, teniendo que crear la matriz etc …

Las pruebas de sistema son las que todo funciona bien, y no haya fallos. Si que es verdad que algunas pruebas no las hemos hecho de forma individual, sino que las hemos hecho en el propio programa completo, haciendo copias para evitar fallos irreversibles. Por ejemplo el añadir los botones a ultima hora.

Por ultimo las pruebas de aceptación será la única parte que esperaremos para hacer, ya que tenemos que ver la manera. En principio Domingo va a ejercer de cliente y vamos a completar este apartado.

En cualquier caso como he comentado antes si vemos alguna cosa mas que sea interesante añadir (Diagramas, imágenes, …) se hará para dejar la memoria lo mejor posible. También quizá deba revisar el resto de correos para revisar que pruebas se han hecho y que fallos ha habido en el programa.

Domingo echara un vistazo al programa cuando pueda ya que ahora esta con exámenes y yo completare los capítulos tanto de diseño como de pruebas para dejar la memoria prácticamente acabada a falta de las conclusiones y los trabajos futuros en los que además de los requisitos que nos quedan por cumplir podríamos pensar en mas simulaciones alternativas (Por ejemplo si te sales cuando lleves un 20% o 30% de las imágenes). De todas formas la idea es dejar la memoria prácticamente lista (suponiendo que el programa este acabado) para que solo quede revisarla de principio a fin.

Recordar también añadir al final de la memoria el enlace de github para poder disponer del código de forma libre desde ahí.

Para acabar recordar que el limite para la entrega de la memoria es el dia 2 de Julio, sabiendo que me tengo que matricular y que van a mirar mas dicha memoria que el programa, por lo que me centrare en esto.

La próxima reunión será cuando este la memoria prácticamente finalizada o si me encuentro alguna duda/error/comentario durante su desarrollo, en cuyos casos contactare con Domingo via correo o whatsapp.