

PLANIFICACIÓN Y GESTIÓN DE PROYECTOS INFORMÁTICOS (2018-2019)
MÁSTER PROFESIONAL DE INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Manual de Coordinación

Alejandro Campoy Nieves
Luis Gallego Quero

9 de noviembre de 2018

Índice

1	Ciclo de vida	3
2	Metodología de desarrollo	3
3	Recursos software desarrollo	3
4	Organización del equipo de trabajo	3
5	Herramientas para comunicaciones en el equipo de trabajo	3
6	Relaciones con el cliente	4
7	Estándares de documentación	4
8	Estándares de código	4
9	Control de versiones	4
10	Gestión de calidad	4

1. Ciclo de vida

El ciclo de vida que vamos a utilizar es iterativo. Se trata de ir obteniendo parte del producto por pequeños bloques a los que se les denominan ciclos de desarrollo o iteraciones. La razón de esto es ir obteniendo desde el primer momento un proyecto funcional. Aunque en las primeras etapas el programa no cumple las expectativas esperadas y no sea fiable dando resultados de calidad dudosa, tenemos la posibilidad de obtener una aplicación útil en la que podemos ir viendo los resultados iniciales, valorarlos y pensar o analizar la estrategia para la siguiente iteración.

Esto es especialmente útil en nuestro proyecto ya que en las primeras iteraciones será muy difícil conseguir un ajuste de parámetros correcto en el algoritmo de Deep Learning, obteniendo aún resultados inferiores a los esperados.

2. Metodología de desarrollo

Ligado al punto anterior, utilizaremos una metodología ágil basada en SCRUM. Cuyo objetivo sea gestionar y planificar el proyecto con posibilidades de cambio a última hora.

3. Recursos software desarrollo

Desarrollaremos este proyecto a través de Spyder, un IDE para Python que nos ayudará al desarrollo de este sistema. Incluyendo las librerías y funcionalidades disponibles que nos permitan tener una base en la que montar este proyecto y que nos aporte una mejora en la calidad y facilidad de trabajo. Para la realización de los distintos diagramas utilizaremos las herramientas que nos proporciona Draw.io, Microsoft Visio y Github como gestor de documentación y controlador de versiones.

4. Organización del equipo de trabajo

La organización que hemos consensuado es de tipo dinámico. Esto quiere decir, que los integrantes de este proyecto desarrollan roles temporales durante un intervalo variable. En resumen, una persona se dedica a supervisar el proyecto principalmente, mientras que la otra lo desarrolla. Estos papeles se intercambian con el objetivo de no saturar las capacidades de trabajo de cada uno.

5. Herramientas para comunicaciones en el equipo de trabajo

Como controlador de versiones utilizaremos Github. Trello para el desarrollo de SCRUM. Y para comunicarnos, principalmente Whatsapp, haciendo uso de Hangouts en contadas ocasiones para posibles videoconferencias.

6. Relaciones con el cliente

1. Entrevista: Primer encuentro con el cliente destinado en esencia al análisis de requisitos del proyecto que abordamos.
2. Reuniones: Siendo la base de SCRUM realizar reuniones periódicas con el cliente, de este modo, podemos presentarle los diferentes avances y, en ese momento, se puede perfilarse distintos aspectos del proyecto con el objetivo de ajustarse a las necesidades del cliente en cuestión.

7. Estándares de documentación

- Como compositor de texto, utilizaremos la herramienta llamada \LaTeX .
- De portada utilizaremos siempre la misma plantilla para darle homogeneidad y continuidad al proyecto.
- Se enumeraran las páginas de forma automática sin contar la portada, de tal forma que se desarrolla un índice acorde a cada parte de la documentación.
- Aunque el texto se escriba en formato \LaTeX , la documentación final del proyecto será presentada en el formato PDF.

8. Estándares de código

- Como vamos a realizar el proyecto en Python, utilizaremos los convenios de presentación típicos de este lenguaje (en las variables, primera letra minúscula y resto de palabras primera letra en mayúscula, por ejemplo).
- Los comentarios serán especificados en castellano siguiendo el formato de comentarios de Python para los distintos módulos.

9. Control de versiones

Tanto para los documentos como el código generado durante el desarrollo de este proyecto, serán almacenados, controlados y dirigidos a través de Github. Esta plataforma nos proporciona un control de versiones ideal para proyectos de este tipo.

10. Gestión de calidad

Los test para realizar las correspondientes pruebas de desarrollo serán elaboradas a través de la herramienta de Python llamada Unittest. También se utilizará como punto de apoyo, las verificaciones y visto bueno por parte del cliente, que será considerado como un punto más del proceso de prueba y testeo.