Proyecto software

Acta reunión

horizontal line

**6 de mayo 2020 / 12:00 AM / Duración: 1h**

# Asistencia

Alberto Calvo, Saul Flores, Alvaro García, Luis García, Alejandro Facorro, Oscar Baselga, Germán Garcés

Profesor: Rubén Béjar.

# Agenda

## **Objetivos de la reunión**

* Comentar la memoria v2

# Acciones tomadas

* Se ha comentado la memoria con las correcciones necesarias

# Objetivos para la siguiente reunión

* Corregir las indicaciones del profesor
* Documentación:
  + 2. Organización del proyecto
    - Nombres en github
  + 3.1.1 Procesos de inicio del proyecto
    - Poner urls repos que hay
  + 3.1.2 Procesos de ejecución y control del proyecto
    - Url discord
    - Definir rol de redactor de actas y asignarlo. Sustituir los nombres por los roles al nombrarlos después
    - Enseñar Métricas del proyecto (Gráficos y tablas) en la versión final.
    - Poner las referencias de los colores en esos colores
    - Cambiar “intentando modularizar” por modularizando basándose en los requisitos y la planificación
  + 3.1.3 Procesos técnicos
    - Quitar versiones concretas pero sí nombrar algunas bibliotecas (las más importantes) y en qué ficheros se encuentran anotadas (Backend: requirements.txt, Android… Web...)
    - README.md Explicar procesos de instalación en el README.md y desde la memoria explicar primero un poco todo y que hay más detalle en el fichero readme.md por posibles cambios
  + 3.2.3 Plan de aseguramiento de la calidad
    - Indicar objetivos de los test (Backend pj toda la API), pensar si poner porcentajes (Backend 100% no? por peticiones de test a toda la API)
  + 3.2.4
    - Alargar los días del diagrama de Gantt, hasta junio
    - Falta la idea de PLANIFICACIÓN de futuro en el diagrama de Gantt, no solo lo que se ha ido haciendo
  + 4.1 Análisis de requisitos
    - No diferenciar RNF entre los de sistema y los de backend y frontend.
    - Asegurarse de que la última versión tiene bien los requisitos con código RF y RNF
  + 4.2 Diseño del sistema
    - Diagrama de despliegue:
      * Angular en el navegador del usuario cuando se hace el despliegue final.
      * Cambiar diagrama de despliegue con la aplicación web del cliente está fuera de Heroku
    - Diagrama de clases de cliente distinto para cada frontend
    - Añadir el patrón arquitectural Singleton y los patrones que estén incluidos en los frameworks
    - **Documentar TODOS los diagramas (explicarlos)**
    - Bibliotecas en los diagramas de paquetes y módulos
    - Añadir clases de **conexión** y **modelo de datos** de Flask a la base de datos (*Domain modeling* en Google)
    - El diagrama de objetos de la BD en el servidor son las tablas
    - Cambiar los diagramas de clases de la BD por el **E/R** e indicar que se está utilizando un ORM (SQLAlchemy)
    - Diagrama de componentes y conectores: profundizar en los clientes indicando ¿qué componentes hacen las peticiones http? Si hay un patrón indicar con un ejemplo (Proxy?). Añadir en Servidor una cajita que ponga modelo de datos (y luego se explica)
    - Documentar la API con un link a la wiki de github
    - Añadir referencias y descripción a las figuras del documento (Alberto Latex)
    - Comentar un poco las interfaces en las imágenes de los prototipos
    - Explicar el mapa de navegación de web y android
      * Añadir un caso de uso para registro
      * Añadir texto o títulos de pantalla (central, registo) o por secciones
  + 5.2 Ejecución y control del proyecto
    - Poner gráficas y métricas del proyecto
    - Quitar el término ‘divergencias’ del calendario del proyecto
  + Memoria global:
    - Corregir referencias
    - Se puede usar la primera persona gramatical para que quede más claro. Es posible decir “Nos decantamos por usar Heroku”.
  + **Enviar apk y url el día de antes de la entrega final**

Próxima reunión

Con profesor: Demostración final 27 de mayo 12:00

Grupo: 10 de mayo 19:00