

## Reunión 13 de febrero de 2018

- Historia de Usuario
  - Roles: Usuario, administrador y mantenimiento (n para cada uno)
    - Usuario:
      - Poder registrarse -> usar app
      - Marcar incidencias -> alguien las solucione
      - Ver incidencias de sala -> no repetir incidencias
      - Recibir notificaciones de incidencias (solucionado, pedir más datos...) -> saber cuando algo se ha resuelto o ha sido rechazado
      - Consultar información del campus en mapa -> interactuar con la aplicación
    - Administrador
      - Ver incidencias filtradas por distintos criterios (lista y mapa) -> poder resolverlas
      - Aceptar o rechazar incidencias (opción de prioridades) -> que la gente de mantenimiento sepa cuáles tiene que resolver
    - Mantenimiento (no hace falta darlos de alta de la propia aplicación)
      - Ver incidencias aceptadas (con horarios de sala) -> asignar incidencias (GUI facilita asignación correcta teniendo en cuenta restricciones)
      - Marcar incidencias como empezadas/completadas -> tener un registro de mi trabajo
      - Poder listar plan de asignación de trabajo -> poder organizarme
- Identificación de usuario no es 100% imprescindible. Se puede dejar abierto o poner registro básico. Se puede permitir también x tiempo de acceso a gente invitada. Posibilidad, e-mail de unizar, enviar correo de confirmación.
- Posibles soluciones problema horarios: Administrador indica horario concreto, hay x personas de mantenimiento
- Proxima semana, convertir a Scrum. Pila de producto. Arquitectura. Posibles ideas GUI.
- Trabajar con puntos de historia
- Tecnologías:
  - Front-end: Web
  - Back-end: Java o Python. Elegir para la semana que viene. Sql Alchemy o Flask
  - Base de datos: recomendable una base de datos. Obligatorio PostgreSQL.