



# **Memoria Sprint 1**

## **Profundización en Ingeniería del Software**

**Escuela Técnica Superior de Ingenieros Informáticos  
Universidad Politécnica de Madrid  
Boadilla del Monte - Madrid  
2016-2017**

**Rampérez Martín, Víctor (MUII)  
Rodríguez Fraga, Alejandro (MUII)  
Sergio Vicente de las Heras (MUII)  
Gabriel García Ardiles (MUII)**

## ÍNDICE

<b>Requisitos especificados</b>	<b>3</b>
<b>Qué se ha entregado</b>	<b>3</b>
<b>Qué no se ha entregado y por qué</b>	<b>4</b>
<b>Gestión del tiempo</b>	<b>4</b>
<b>Problemas encontrados</b>	<b>4</b>
<b>Lecciones aprendidas</b>	<b>5</b>

# 1.Requisitos especificados

Se nos ha pedido el desarrollo de una aplicación web para poner en contacto a personas de una ciudad mediante la publicación y suscripción a actividades o eventos. Por ejemplo, un grupo de estudiantes de intercambio acaban de llegar a su nueva ciudad y quieren conocer gente. Para ello podrán buscar en la aplicación eventos que previamente otros usuarios han publicado (e.g. quedar en un bar a tomar algo, ir al cine, jugar un partido, etc) y marcar su asistencia.

Más concretamente los requisitos especificados para este primer sprint son:

- Permitir el registro de nuevos usuarios. Para ello el usuario proporcionará los siguientes datos: nombre, contraseña, email, género, edad y foto de perfil, su ciudad.
- Permitir el acceso de los usuarios mediante sus credenciales.
- Permitir a los usuarios registrados crear nuevas actividades. Para ello se deberá proporcionar la siguiente información: título del evento, breve descripción, ciudad, lugar en la ciudad (opcional), fecha, hora y fotografía (opcional). El usuario solo podrá crear eventos en su propia ciudad.
- Permitir listar todas las actividades y filtrar dichas actividades por una ciudad en concreto.
- Estas funcionalidades solo deberán estar implementadas a nivel de API (sin desarrollar un capa de frontend).

## 2.Qué se ha entregado

- Permitir el acceso de los usuarios mediante sus credenciales.
- Base de datos SQLite que almacena: ciudades disponibles, información de los eventos creados, información de los usuarios.
- Backend API en NODE.JS que permite:
  - Registro de nuevos usuarios.
  - Login de usuarios en la aplicación.
  - Creación de nuevas actividades.
  - Búsqueda de actividades y filtrado por ciudad.
- Testing de la base de datos y del api del backend utilizando mocha.JS.

### 3. Qué no se ha entregado y por qué

Se han entregado todas las funcionalidades requeridas. No obstante, nos ha faltado algo de tiempo para realizar tests más exhaustivos. La principal razón para que no se haya realizado ha sido la falta de tiempo, debido a que empezamos el sprint algo tarde (tuvimos que aplazar la primera reunión por temas de compatibilidades de horarios) y a la falta de conocimiento de las tecnologías usadas por parte de algunos miembros del equipo.

### 4. Gestión del tiempo

En este primer sprint los requisitos para el desarrollo han sido divididos de manera que todos los integrantes hemos tenido unas responsabilidades generales, como es el diseño general del sistema, y además responsabilidades específicas dentro del proyecto.

Las tareas generales incluyen:

- Elección de la tecnología a usar para el servidor: NodeJS.
- Elección de la base de datos inicial: SQLite.
- Framework de testing: mochaJS.
- No usar ORMs para no perder flexibilidad en nuestras queries a la base de datos.

Los integrantes del grupo además hemos tenido las siguientes responsabilidades concretas:

1. Víctor: Desarrollo de la API ofrecida por el servidor.
2. Alejandro: Desarrollo de funciones encargadas de la interacción con la base de datos.
3. Gabriel: Testing de los métodos que consultan y modifican la DB
4. Sergio: Testing de la API

### 5. Problemas encontrados

Algunos de los problemas que nos hemos tenido que enfrentar en este primer sprint han sido:

- Conseguir buscar una fecha que nos permitiese a todos poder tener la reunión inicial con la gente de especificación de los requisitos. Debido a motivos de trabajo y clases nos fue bastante complicado acordar una fecha para la reunión. Dicha reunión la tuvimos que posponer casi una semana y

esto nos obligó a correr un poco más en el desarrollo en los últimos días del sprint.

- No poder usar un control de versiones. Debido a que en este primer sprint no podíamos usar una herramienta para el control de versiones, tuvimos que compartir el código mediante otras herramientas (e.g. dropbox, email), lo cual nos ha dificultado enormemente el poder trabajar de forma individual, ya que requería constante comunicación entre los miembros del equipo.
- Desconocimiento de las tecnologías utilizadas por algunos miembros del equipo. Una parte del tiempo ha sido empleado en el estudio de estas tecnologías, lo que no produce nada entregable pero es necesario para la realización del código final.

## 6. Lecciones aprendidas

Relacionado con los problemas descritos en la sección previa, hemos extraído las siguientes conclusiones:

- Importancia de una herramienta de control de versiones. Estamos tan acostumbrados al uso de dichas herramientas (e.g. git), que al tener que enfrentarnos al desarrollo del proyecto sin dichas herramientas nos ha permitido comprender la importancia crucial que estas tienen en el desarrollo software.
- Comunicación entre miembros del equipo. Debido a la falta de tiempo de los miembros del equipo para poder tener reuniones presenciales con frecuencia, hemos aprendido a comunicarnos de forma más efectiva mediante otros medios (e.g. whatsapp, email, etc) y a una distribución más eficiente del trabajo entre los miembros del equipo.
- Ha sido crucial definir los contratos básicos, utilizados por cada uno de los integrantes del equipo de desarrollo, que nos permitan definir las entradas y salidas requeridas por los métodos implementados y así poder trabajar conjuntamente en equipo sin tener que hacer un desarrollo secuencial, es decir, sin tener que esperar a que las tareas de los demás integrantes estén finalizadas antes de poder desarrollar las propias.