



# Memoria Sprint 2

## Profundización en Ingeniería del Software

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

**Repositorio:**

**<https://github.com/Mognom/Prof.-Ing.-Software.git>**

**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>3</b>
<b>Estimación/Gestión del tiempo</b>	<b>3</b>
<b>Problemas encontrados</b>	<b>4</b>
<b>Lecciones aprendidas</b>	<b>4</b>

# 1.Requisitos especificados

En este segundo sprint se nos ha pedido que implementemos toda la interfaz gráfica de la aplicación responsive, así como aumentar algunas de las funcionalidades de la API desarrolladas en el sprint anterior. Más concretamente se nos ha pedido:

- Interfaz Gráfica para el registro de usuarios. Debe ser responsive.
- Interfaz gráfica para el login de usuarios (local). Debe ser responsive.
- Interfaz gráfica para creación de eventos responsive.
- Interfaz gráfica para listar eventos responsive.
- Adaptar la aplicación para permitir futuramente la autenticación basada en redes sociales.

## 2.Qué se ha entregado

Se han conseguido entregar todos los requisitos especificados para este sprint. Cabe destacar que han sido necesarias algunas hora extras para poder completarlo completamente. Se han entregado los siguientes requisitos:

- Interfaz Gráfica para el registro de usuarios. Debe ser responsive.
- Interfaz gráfica para el login de usuarios (local). Debe ser responsive.
- Interfaz gráfica para creación de eventos responsive.
- Interfaz gráfica para listar eventos responsive.
- Adaptar la aplicación para permitir futuramente la autenticación basada en redes sociales.

## 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 de toda la interfaz gráfica, debido principalmente a que desconocíamos los frameworks existentes.

## 4.Estimación/Gestión del tiempo

El reparto del trabajo ha sido el siguiente:

1. Víctor: Soporte en el desarrollo de la interfaz gráfica y generación de documentación (tanto de la aplicación como para entregar en la asignatura).
2. Alejandro: Desarrollo de interfaz gráfica.

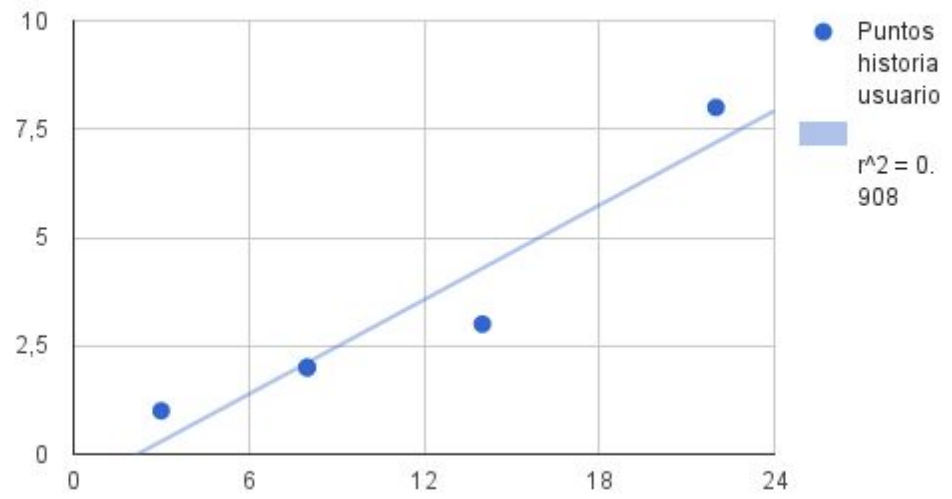
3. Gabriel: Testing.
4. Sergio: Preparar el back-end para permitir la autenticación con redes sociales.

A continuación se muestra una tabla con las horas ideales, horas reales y puntos de historia de usuario para este segundo sprint:

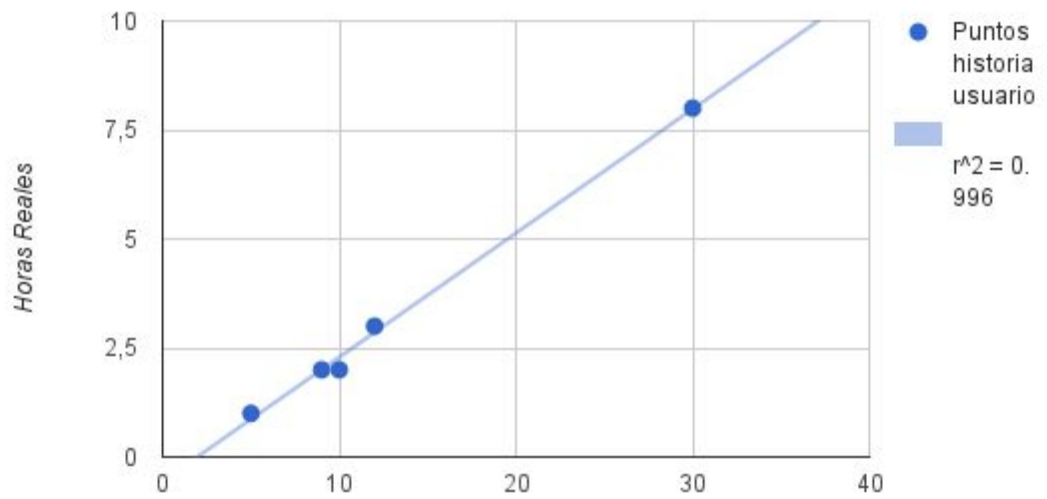
# Feature	Feature	Horas Reales	Horas Ideales	Puntos historia usuario
1	Interfaz Gráfica para el registro de usuarios responsive	12	14	3
2	Interfaz gráfica para el login de usuarios (local) responsive	9	8	2
3	Adaptar la aplicación para permitir futuramente la autenticación basada en redes sociales.	30	22	8
4	Interfaz gráfica para creación de eventos responsive	10	8	2
5	Interfaz gráfica para listar eventos responsive	5	3	1
	TOTAL	66	55	

A partir de la tabla anterior se han generado las siguientes gráficas:

**Horas Ideales - Puntos historia usuario**



**Horas Reales - Puntos historia de usuario**





## 5. Problemas encontrados

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

- El principal problema que hemos encontrado este sprint ha sido la especificación de los requisitos para este segundo sprint, ya que han sido muy vagas las explicaciones y contradictorias con respecto a las proporcionadas por otro componente del equipo de requisitos en el Sprint 1. Esto nos ha obligado a implementar la parte gráfica (requisito principal de este segundo sprint), teniendo que implementar otras funcionalidades a nivel de API que no se nos han requerido para este sprint, pero que se nos sugirieron que formarían parte de este Sprint en la reunión inicial del proyecto (e.g. autenticación basada en redes sociales).
- Dificultad a la hora de establecer el punto de conexión entre los desarrolladores del *front-end* y *back-end*, ya que cambios a nivel de *back-end* producían fallos en el código previamente desarrollado en *front-end* (dificultad para sincronizar ambos desarrollos de forma paralela).

## 6. Lecciones aprendidas

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

- Necesidad de establecer mediante un modelo formal los puntos de conexión entre el desarrollo *back-end* y *front-end* para poder paralelizar ambos desarrollos. Además, es interesante la programación de tests para observar y reparar los fallos que produce un cambio en alguno de esos desarrollos (e.g. un cambio en una función del *back-end* produce un error a la hora de mostrar la información por el *front-end*).