



# Memoria Sprint 3

## 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>Descomposición en slices</b>	<b>3</b>
<b>Qué se ha entregado</b>	<b>4</b>
<b>Qué no se ha entregado y por qué</b>	<b>5</b>
<b>Estimación/Gestión del tiempo</b>	<b>5</b>
<b>Problemas encontrados</b>	<b>7</b>
<b>Lecciones aprendidas</b>	<b>8</b>
<b>Otras conclusiones</b>	<b>9</b>

# 1.Requisitos especificados

Para este último sprint, además de los requisitos correspondientes, se nos ha notificado un cambio con respecto al desarrollo de la aplicación original. Los requisitos especificados para este sprint han sido los siguientes:

- Ver perfil de usuario (API + interfaz gráfica).
- Editar el perfil del usuario (API + interfaz gráfica).
- Poder suscribirse a los distintos eventos (API + interfaz gráfica).
- Poder escribir comentarios en los eventos (API + interfaz gráfica).
- Filtrar eventos por fecha (API + interfaz gráfica).
- Filtrar eventos por usuario de creación (API + interfaz gráfica).
- Poder enviar mensaje entre los usuarios\* (API + interfaz gráfica).

\*Este ha sido el requisito que hace que tengamos que modificar parte del desarrollo realizado hasta ahora.

# 2.Descomposición en slices

A continuación se muestra la descomposición en slices que se ha realizado de los requisitos planteados para este último sprint:

Requisito	Slices
Ver el perfil de usuario	Servicio REST a nivel de API que devuelve la información del usuario
	Opción en la interfaz para ver el perfil
	Página html para mostrar la información del usuario
Editar el perfil de usuario	Servicio REST a nivel de API que modifica la información del usuario
	Opción en la interfaz para editar el perfil
	Página html para modificar la información del usuario
Poder suscribirse a eventos	Servicio REST a nivel de API añadir un usuario a un evento
	Servicio REST a nivel de API para obtener todos los usuarios que participan en un evento

	Modificar la página html para poder mostrar los usuario inscritos en cada evento
Poder escribir comentarios en evento	Servicio REST a nivel de API para añadir un comentario a un evento
	Añadir opciones para escribir y editar un comentario en la página html donde se muestra un evento
Filtrar eventos por fecha	Modificar servicio REST para obtener eventos
	Añadir parámetro de filtrado por fecha en la página html que muestra los eventos
Filtrar eventos por creador	Modificar servicio REST para obtener eventos
	Añadir parámetro de filtrado por fecha en la página html que muestra los eventos
Envío de mensajes entre usuarios	Crear servicio en backend para registrar mensajes
	Crear servicio para la distribución de los mensajes
	Crear servicio de notificación de mensajes a los usuarios
	Crear opción de chat en la barra de navegación
	Diseño e implementación de la página html del chat

En este caso las slices son quizás más gruesas de los general debido a el proyecto se encuentra bastante desarrollado y por tanto la gran mayoría de requisitos son pequeñas modificaciones. Nótese que por ejemplo las slices son más finas para el requisito de los mensajes entre usuario al tratarse de un requisito novedoso y no previsto.

### 3. Qué se ha entregado

Al finalizar este sprint se ha entregado la implementación de todos los requisitos especificados dentro del plazo de dicho sprint y sin superar las horas acordadas para el desarrollo de este proyecto dentro de la asignatura. Concretamente se ha entregado los siguientes requisitos:

- Ver perfil de un usuario (API + interfaz gráfica).
- Editar el perfil del usuario (API + interfaz gráfica).
- Poder suscribirse a los distintos eventos (API + interfaz gráfica).
- Poder escribir comentarios en los eventos (API + interfaz gráfica).
- Filtrar eventos por fecha (API + interfaz gráfica).
- Filtrar eventos por usuario de creación (API + interfaz gráfica).

- Poder enviar mensaje entre los usuarios\* (API + interfaz gráfica).

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

Aunque, como se ha mencionado en la anterior sección, se han podido implementar todos los requisitos de este sprint, es destacable que nos hubiese gustado poder disponer de algo más de tiempo para poder realizar tests de esta parte (no se ha realizado ningún test en este sprint) y poder dedicar más tiempo a mejorar aspectos de visuales de usabilidad de la interfaz gráfica. Creemos que esta falta de tiempo (unas pocas horas hubieran sido suficientes) se ha debido al cambio de requisitos (mensajes entre usuarios), que al no tenerlo previsto desde el comienzo ha provocado tener que invertir bastante tiempo para encontrar una solución que se pudiese integrar fácilmente en la aplicación.

## 5. Estimación/Gestión del tiempo

El reparto del trabajo ha sido el siguiente:

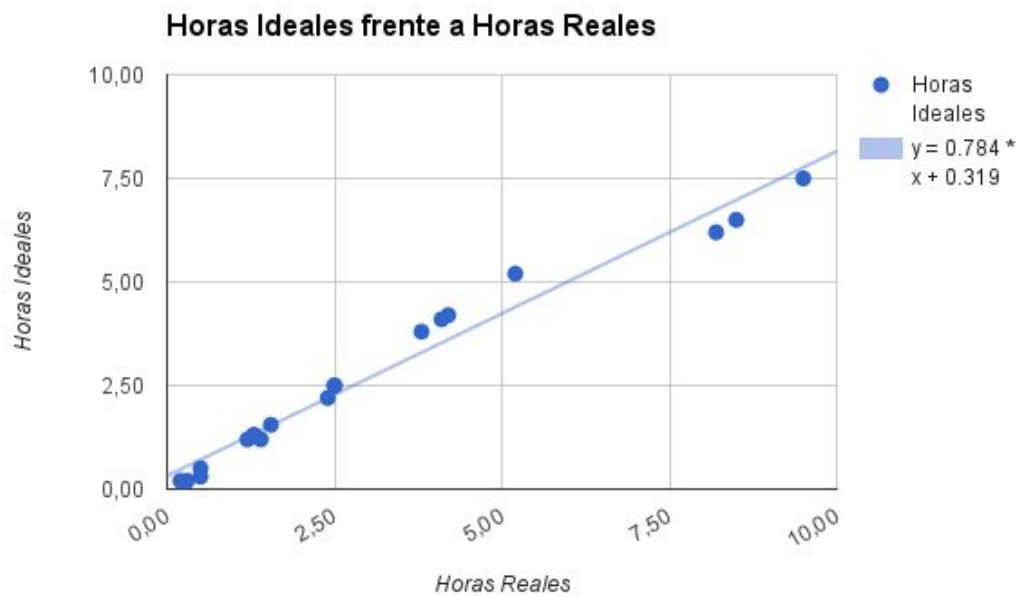
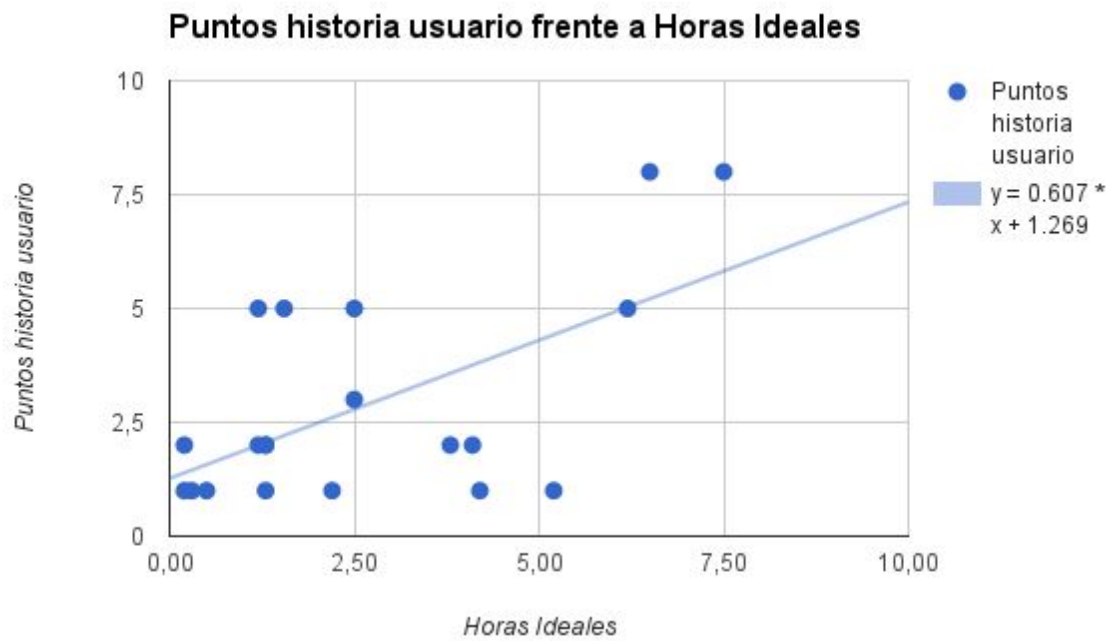
1. Víctor: Soporte en el desarrollo de los mensajes entre usuarios y generación de documentación (tanto de la aplicación como para entregar en la asignatura).
2. Alejandro: Desarrollo de los mensajes entre usuarios.
3. Gabriel: Vista y edición del perfil de usuarios.
4. Sergio: Suscripción a eventos, comentarios en eventos y filtrado de eventos.

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

#	Feature	Slice	Horas Reales	Horas Ideales	Puntos historia usuario
1	Ver el perfil de usuario	Servicio REST a nivel de API que devuelve la información del usuario	2,50	2,50	3
		Opción en la interfaz para ver el perfil	0,30	0,20	2
		Página html para mostrar la información del usuario	1,30	1,30	1
2	Editar el perfil de usuario	Servicio REST a nivel de API que modifica la información del usuario	1,55	1,55	5
		Opción en la interfaz para editar el perfil	0,50	0,30	1

		Página html para modificar la información del usuario	1,30	1,30	2
3	Poder suscribirse a eventos	Servicio REST a nivel de API añadir un usuario a un evento	2,50	2,50	5
		Servicio REST a nivel de API para obtener todos los usuarios que participan en un evento	3,80	3,80	2
		Modificar la página html para poder mostrar los usuario inscritos en cada evento	2,40	2,20	1
4	Poder escribir comentarios en evento	Servicio REST a nivel de API para añadir un comentario a un evento	1,40	1,20	5
		Añadir opciones para escribir y editar un comentario en la página html donde se muestra un evento	1,20	1,20	2
5	Filtrar eventos por fecha	Modificar servicio REST para obtener eventos	1,30	1,30	2
		Añadir parámetro de filtrado por fecha en la página html que muestra los eventos	0,50	0,50	1
6	Filtrar eventos por creador	Modificar servicio REST para obtener eventos	4,10	4,10	2
		Añadir parámetro de filtrado por fecha en la página html que muestra los eventos	4,20	4,20	1
7	Envío de mensajes entre usuarios	Crear servicio en backend para registrar mensajes	8,50	6,50	8
		Crear servicio para la distribución de los mensajes	8,20	6,20	5
		Crear servicio de notificación de mensajes a los usuarios	9,50	7,50	8
		Crear opción de chat en la barra de navegación	0,20	0,20	1
		Diseño e implementación de 1 página html del chat	5,20	5,20	1
	<b>TOTAL</b>		60,45	53,75	

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



## 6. Problemas encontrados

Este sprint ha sido el menos problemático a la hora de gestionar el trabajo entre los miembros y la comunicación entre los mismos, probablemente debido a la experiencia adquirida en los dos sprints previos. El único problema destacable que nos ha surgido ha sido el cambio en la especificación de requisitos, que aunque no altera gravemente las funcionalidades anteriormente implementadas, ha provocado el tener que buscar una solución lo suficientemente flexible para poder ser

introducida en la aplicación sin haber proyectado su inclusión desde el principio del proyecto. No obstante, somos consciente de que esta situación es bastante común en la gran mayoría de desarrollos software y se ha podido resolver, eso sí provocando una desviación entre el tiempo estimado y el realmente utilizado para esa tarea.

## 7. Lecciones aprendidas

Al ser este el último documento del proyecto, nos gustaría que esta sección reflejase las lecciones aprendidas, no solo de este último sprint sino del conjunto de todo el proyecto, ahora que tenemos una visión global del mismo:

- Dificultad para estimar el tiempo de desarrollo en etapas tempranas del desarrollo software (en el *sprint 1*). Sobre todo si la experiencia con las herramientas es irregular entre los miembros del grupo. Esto puede provocar desviaciones entre los tiempos estimados y los tiempos reales de desarrollo muy importantes como nos sucedió en el *sprint 1*.
- Dificultad para definir los puntos de unión del software en un desarrollo distribuido. Nos tuvimos que enfrentar en el primer a sprint a la definición de las “fronteras” entre las distintas partes del software que implementaría cada miembro de forma individual para no perder excesivo tiempo a la hora de unirlo todo debido a fallos en la definición. Observamos cómo esta definición en más fácil a medida que el equipo gana experiencia tanto en el proyecto como con la forma de trabajar del resto de miembros del equipo.
- Dificultad para trabajar sin una herramienta de control de versiones y/o repositorio común. Esto lo sufrimos en el sprint1 y obligó a trabajar de forma secuencial, lo cual ralentizaba todo el desarrollo mucho.
- Dificultad para estimar el tiempo de desarrollo de los cambios de requisitos. Esto nos ha sucedido en este último sprint, pues al considerar que el cambio de requisitos no modificaba funcionalidades ya implementadas, estimamos muy poco tiempo para su implementación. Finalmente, comprobamos como nos obligaba a modificar algunos detalles de funcionalidades anteriores, lo cual provocó desviaciones entre los tiempos estimados y reales. No obstante, creemos que estas desviaciones cuando hay cambios de requisitos son bastante comunes, ya que es difícil a simple vista ver los cambios que va a conllevar asociado un cambio en los requisitos.
- Hemos sido testigo de una mala práctica bastante habitual como es no programar tests (automáticos) en los sprints en los que el tiempo es más limitado (los dos últimos sprints en nuestro caso). No obstante, no hemos comprobado los beneficios de tener implementados dichos tests, ya que apenas hemos tenido que retocar el código anteriormente implementado (sólo en el último sprint). Estas modificaciones hicieron que dejaran de funcionar algunas de las funcionalidades implementadas en otros sprints y no fuimos conscientes de ello hasta que volvimos a probar todo el sistema. Sin



embargo, si hubiésemos tenido tests automáticos, podríamos haber visto enseguida que al modificar una parte estábamos teniendo efectos colaterales.

## 8. Otras conclusiones

Por último, y al tratarse del último documento, nos gustaría dejar constancia de algunos comentarios personales acerca de la asignatura y el proyecto:

- Consideramos que hubiese sido interesante tratar algunos temas como integración continua (sabemos que no se ha impartido por falta de tiempo).
- Quizás la parte referente a desarrollo Agile debería estar más presente debido a que es una metodología de desarrollo ampliamente utilizada y cada vez más.
- Profundizar en algunos aspectos de testing, como test de integración y sistema de forma avanzada.