

# Gestión del proyecto

Diego Martín Arroyo

Miércoles, 8 de julio de 2015

## Índice

<b>Introducción</b>	<b>1</b>
<b>Planificación</b>	<b>1</b>
Definición de la metodología . . . . .	2
Herramientas de planificación . . . . .	2
Planificación temporal . . . . .	3
Diagrama de Gannt (parte 1) . . . . .	7
Diagrama de Gannt (parte 2) . . . . .	8
Diagrama de Gannt (parte 3) . . . . .	9
Diagrama de Gannt (parte 4) . . . . .	10
Diagrama de Gannt (parte 5) . . . . .	11
Diagrama de Gannt (parte 6) . . . . .	12
<b>Gestión de la calidad</b>	<b>13</b>
<b>Comunicación</b>	<b>13</b>
<b>Análisis de riesgos</b>	<b>13</b>
<b>Control de versiones</b>	<b>13</b>

## Introducción

El presente documento tiene como objetivo describir todos los aspectos relativos a la gestión del proyecto no detallados en la memoria principal del mismo, definiendo las secciones del mismo siguiendo las indicaciones del *Project Management Body of Knowledge (PMBOK)*.

## Planificación

Uno de los aspectos fundamentales a la hora de realizar cualquier proyecto de desarrollo de una envergadura considerable es la planificación del mismo. Esta necesidad es mayor acuciante cuando se consideran diferentes restricciones como las dadas en el presente proyecto, generalmente de carácter temporal. Otro aspecto a considerar en este proyecto a la hora de elegir la metodología a utilizar es la falta de experiencia con el tipo de *software* a desarrollar. La incertidumbre es muy alta, y por tanto es crucial elegir una dinámica de desarrollo tolerante a cambios inesperados y “camino sin salida”.

La metodología por tanto debe ser elegida cuidadosamente, basándose en modelos probados y conocidos, adaptando si es necesario estos a las necesidades específicas del proyecto.

## Definición de la metodología

Tras varias reuniones en las que se establecen los objetivos globales del proyecto, realizadas en septiembre de 2014 y a partir de febrero de 2015 de forma más intensiva, se decide en marzo, previamente a que comience de forma intensiva el desarrollo del proyecto el uso de una metodología ágil. Esta se basará en la realización de etapas de trabajo de corta duración (entre una y dos semanas). Con reuniones de equipo entre cada una de ellas. En estas reuniones se evalúan los resultados, identifican situaciones críticas y se definen los objetivos de la siguiente etapa.

A fin de dotar al proyecto de la flexibilidad necesaria para subsanar la falta de experiencia en diferentes componentes del proyecto, se opta por basar los diferentes desarrollos en un proceso orientado a prototipos incrementales, productos *software* incompletos que incorporan diferentes partes de la funcionalidad a desarrollar y que son evaluables. De esta forma, es posible validar la viabilidad de un producto sin que el mismo esté completo, realizar un análisis de alternativas detallado (elaborando prototipos orientativos en poco tiempo) del mismo y añadir nuevas mejoras de forma incremental (de esta forma, en caso de que no todos los objetivos marcados puedan marcarse, el resto del producto será funcional).

Esta metodología ha permitido la ampliación de los objetivos del proyecto sin alterar el ritmo de desarrollo. El desarrollo de herramientas como **MarcoPolo**, hoy parte crucial del sistema e inicialmente no planteada, ha sido factible gracias a la elección de este proceso.

En cada una de las iteraciones se llevan a cabo las actividades de iniciación, planificación, ejecución, control y conclusión habituales en todo proyecto.

## Herramientas de planificación

Se considera que el uso de una herramienta para la gestión del proyecto es beneficiosa. Su utilización facilita la identificación de tareas a realizar, coordinar el equipo y obtener de forma sencilla una visualización del estado del proyecto.

Se valoran varias herramientas con las que se cuenta experiencia o conocidas. **Microsoft Project**<sup>1</sup>, **JIRA**<sup>2</sup> y **Redmine**<sup>3</sup>. Microsoft Project es descartado por la falta de funcionalidad en línea (a menos que se utilice Microsoft Project Server, cuya adquisición no es considerada. El hecho de contar con una instancia en línea permite coordinar un equipo que no se encuentra en la misma localización física y el acceso a la plataforma desde cualquier dispositivo (gran parte de la planificación se ha realizado a través de un cliente móvil para la plataforma Android). En el caso particular de este proyecto, como Trabajo de Fin de Grado, el uso de este tipo de instancia permite a los tutores contar con un seguimiento del alumno sin la necesidad de reuniones y a los evaluadores del mismo la revisión de la gestión del proyecto de forma sencilla.

**JIRA** es descartado al no contar con experiencia de uso y su coste, previsiblemente alto (así como los costes de alojamiento). **Redmine** es una herramienta libre y gratuita, con la que se cuenta experiencia, satisface todas las demandas planteadas y su consumo es pequeño. Es por ello que es la alternativa a utilizar en el proyecto.

Redmine define todas las tareas del proyecto en “asuntos” *issues*, que pueden ser de tres tipos: error (*bug*), funcionalidad (*feature*) o soporte (*support*). Estos asuntos pueden ser marcados temporalmente, incluyendo estimaciones de tiempo, progreso, notas y comentarios. Todos los datos introducidos a la plataforma serán representados en un diagrama de Gantt que representará el estado de desarrollo del proyecto.

La instancia de Redmine utilizada se encuentra alojada en <http://redmine.martinarroyo.net/>. El acceso a la misma no requiere autenticación.

---

<sup>1</sup><https://products.office.com/es-ES/project>

<sup>2</sup><https://www.atlassian.com/software/jira>

<sup>3</sup><http://redmine.org/>

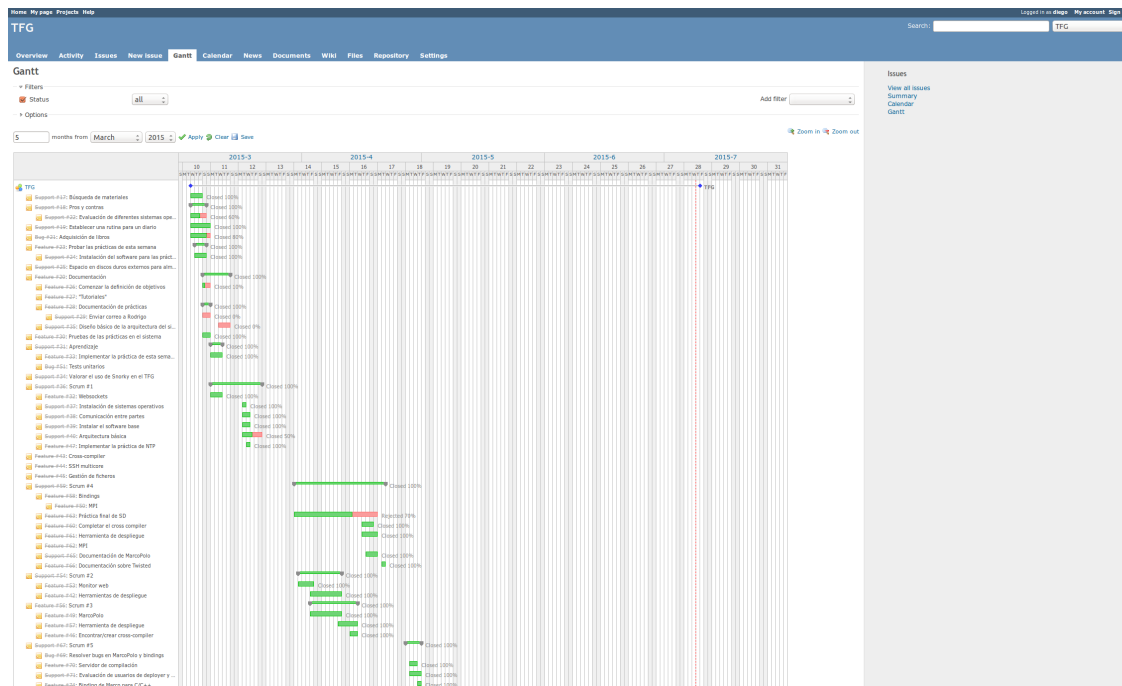


Figura 1: Vista parcial de todos los asuntos del proyecto en Redmine

## Planificación temporal

En total se han realizado un total de 13 iteraciones en un periodo de 16 semanas, marcadas por reuniones de inicio y fin en cada una de ellas. A continuación se detalla brevemente las características de cada una de ellas:

### Iteración 1

- **Fecha de inicio:** 09/03/2015
- **Fecha de fin:** 21/03/2015
- En esta iteración comienza formalmente el desarrollo del proyecto.
- **URL:** <http://redmine.martinarroyo.net/issues/36>

### Iteración 2

- **Fecha de inicio:** 31/03/2015
- **Fecha de fin:** 10/04/2015
- Los esfuerzos se concentran en las herramientas de despliegue y el monitor web.
- **URL:** <http://redmine.martinarroyo.net/issues/54>

**Iteración 3**

- **Fecha de inicio:** 03/04/2015
- **Fecha de fin:** 14/04/2015
- Comienza el desarrollo del protocolo de descubrimiento de servicios **MarcoPolo**, la creación de un compilador cruzado y la herramienta de despliegue. Esta iteración continúa el trabajo llevado a cabo en la iteración 2.
- **URL:** <http://redmine.martinarroyo.net/issues/56>

**Iteración 4**

- **Fecha de inicio:** 30/03/2015
- **Fecha de fin:** 21/04/2015
- Comienza el desarrollo de los *Bindings* de **MarcoPolo**, se define el compilador cruzado y MPI es incluido en el proyecto
- **URL:** <http://redmine.martinarroyo.net/issues/59>

**Iteración 5**

- **Fecha de inicio:** 27/04/2015
- **Fecha de fin:** 30/04/2015
- Se establece un servidor de compilación y la expansión del número de bindings comienza, así como diferentes procesos de evaluación de usuarios.
- **URL:** <http://redmine.martinarroyo.net/issues/67>

**Iteración 6**

- **Fecha de inicio:** 30/04/2015
- **Fecha de fin:** 13/05/2015
- Comienza el desarrollo de las herramientas **PoloUsers**, **Marcobootstrap** y **marco-netinst** y se integra el servidor LDAP.
- **URL:** <http://redmine.martinarroyo.net/issues/77>

**Iteración 7**

- **Fecha de inicio:** 11/05/2015
- **Fecha de fin:** 17/05/2015
- En esta iteración se realizan tareas de recopilación y documentación.
- **URL:** <http://redmine.martinarroyo.net/issues/86>

### Iteración 8

- **Fecha de inicio:** 18/05/2015
- **Fecha de fin:** 24/05/2015
- Se desarrollan las versiones iniciales de **Marcollogger** y **marco-bootstrap-backend** (entonces conocido como **marco-bootstrap-backend**).
- **URL:** <http://redmine.martinarroyo.net/issues/89>

### Iteración 9

- **Fecha de inicio:** 09/05/2015
- **Fecha de fin:** 07/06/2015
- Comienza el desarrollo de la estructura física, el *binding* para Polo (hasta ahora se contaba únicamente con el de Marco) y la herramienta **marcomanager**. Debido a que la tarea #68 (Estructura física) debe extenderse durante un mes, la fecha de finalización de la interacción es alterada significativamente de la fecha en la que todos los objetivos finalizaron.
- **URL:** <http://redmine.martinarroyo.net/issues/97>

### Iteración 10

- **Fecha de inicio:** 30/05/2015
- **Fecha de fin:** 07/06/2015
- Ejecución de comandos en Deployer, API de LEDs, documentación, cableado físico y pruebas en entorno real.
- **URL:** <http://redmine.martinarroyo.net/issues/101>

### Iteración 11

- **Fecha de inicio:** 08/06/2015
- **Fecha de fin:** 29/06/2015
- Se intenta añadir soporte a Hadoop para el proyecto, se crean los primeros paquetes instalables, y se da soporte completo a C, C++ y Java con MarcoPolo. Se experimenta además con Syslog.

### Iteración 12

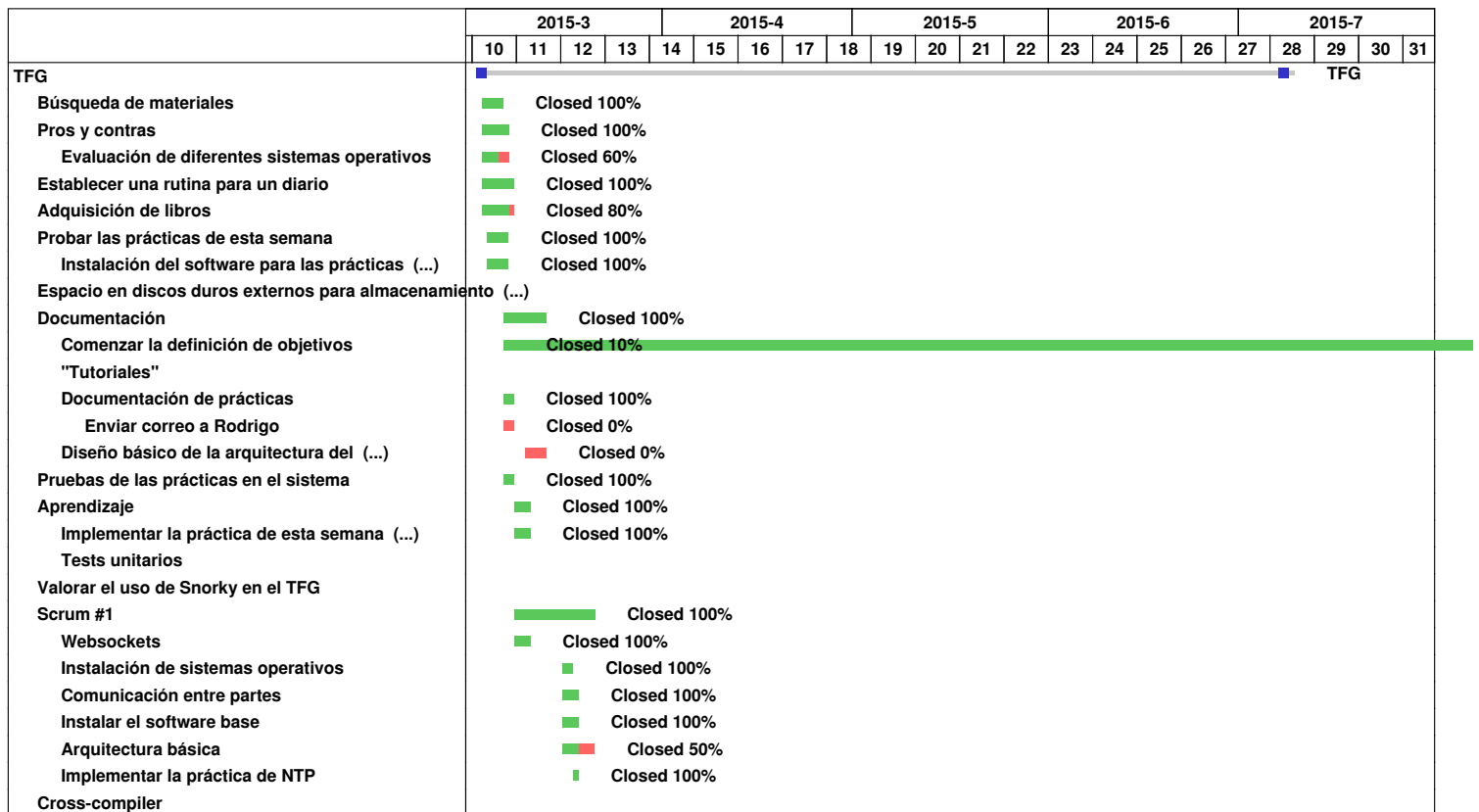
- **Fecha de inicio:** 15/06/2015
- **Fecha de fin:** 26/06/2015
- Características de seguridad en Polo, evaluaciones de usuario y gestión de paquetes.
- **URL:** <http://redmine.martinarroyo.net/issues/118>

**Iteración 13**

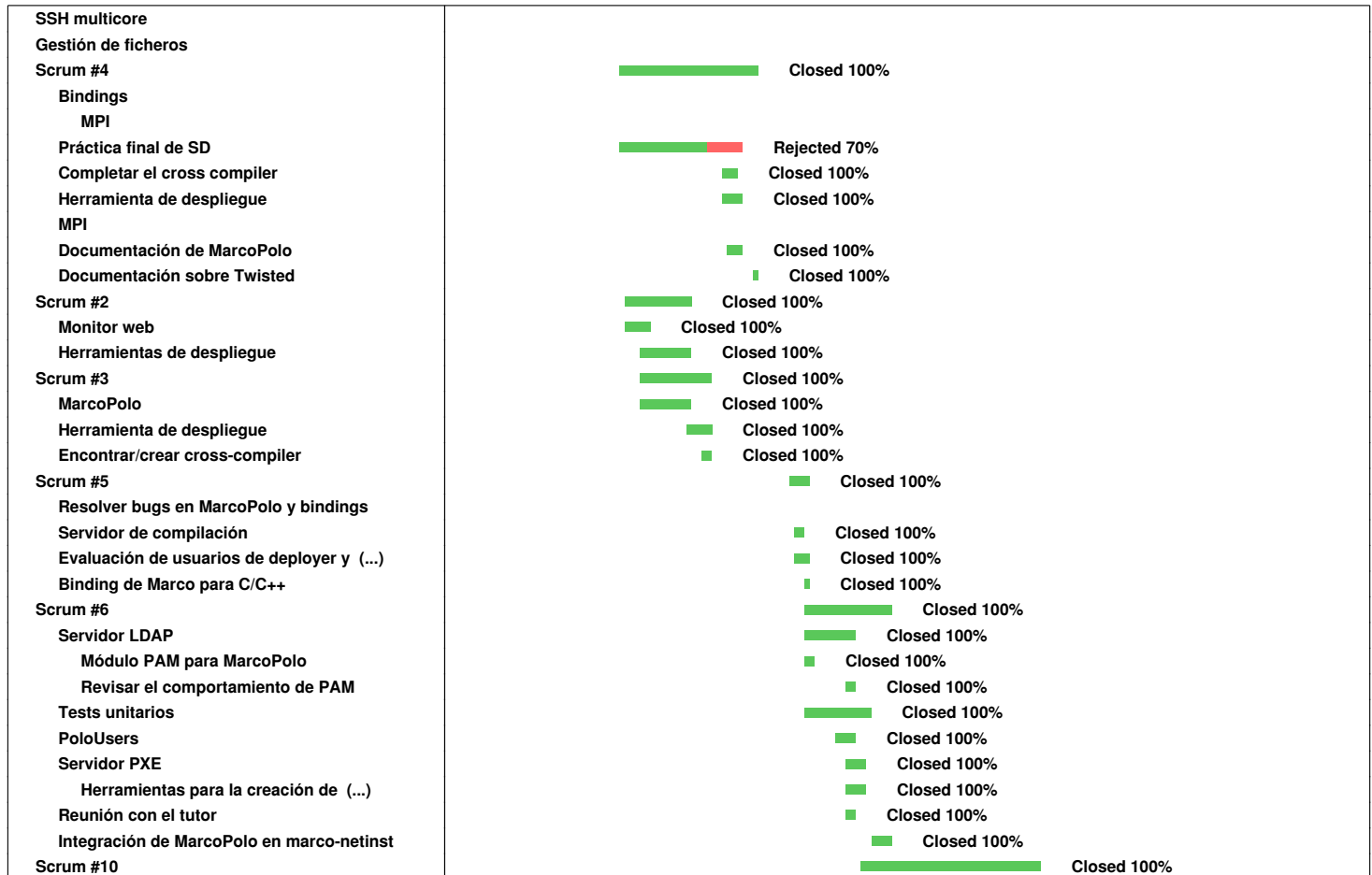
- **Fecha de inicio:** 06/06/2015
- **Fecha de fin:** 09/07/2015
- Esta última iteración reúne todas las tareas pospuestas anteriormente, y por tanto su extensión temporal es mayor.
- **URL:** <http://redmine.martinarroyo.net/issues/148>

## Diagrama de Gannt (parte 1)

## TFG

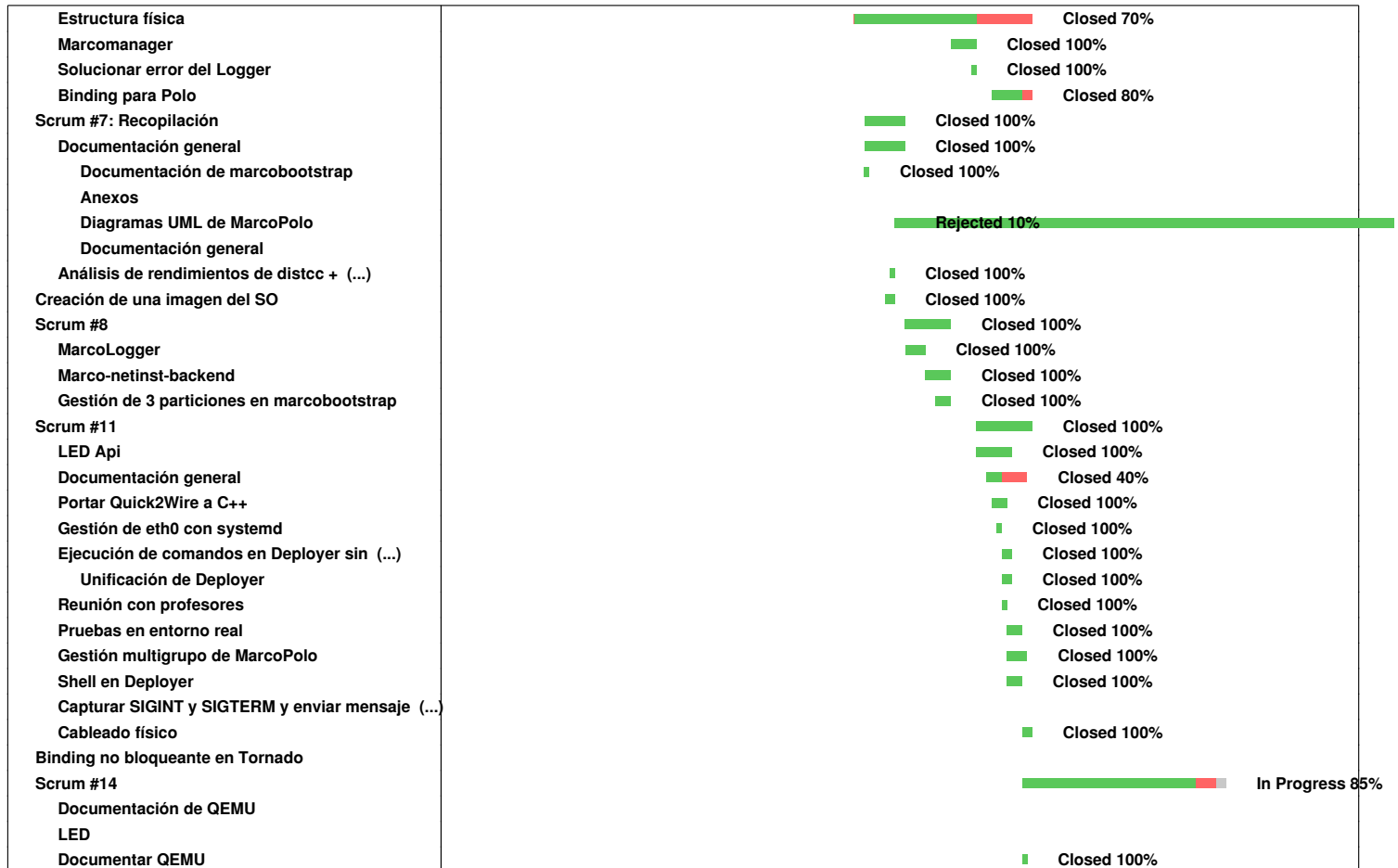


## Diagrama de Gannt (parte 2)

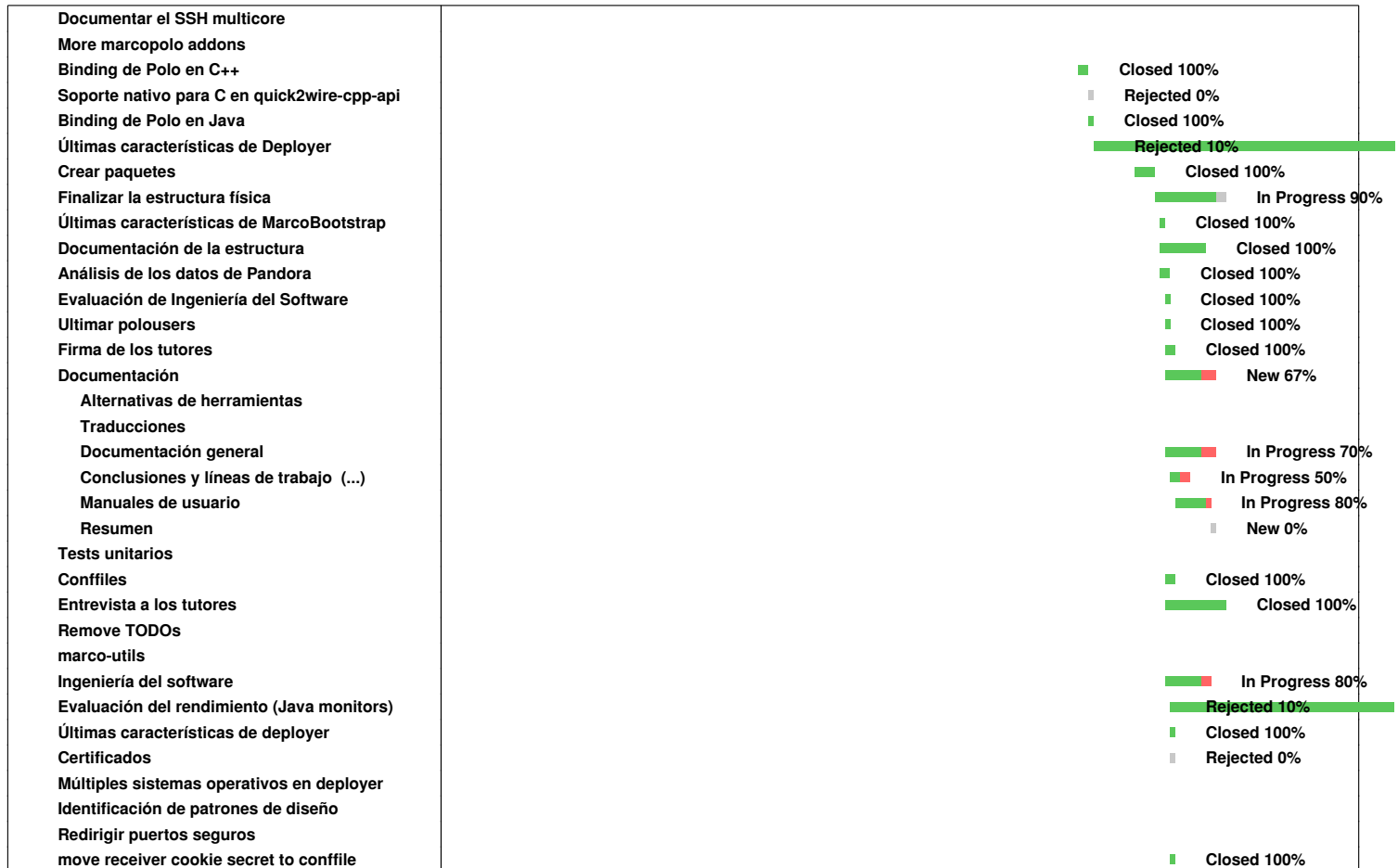




## Diagrama de Gannt (parte 3)



## Diagrama de Gannt (parte 4)



## Diagrama de Gannt (parte 5)

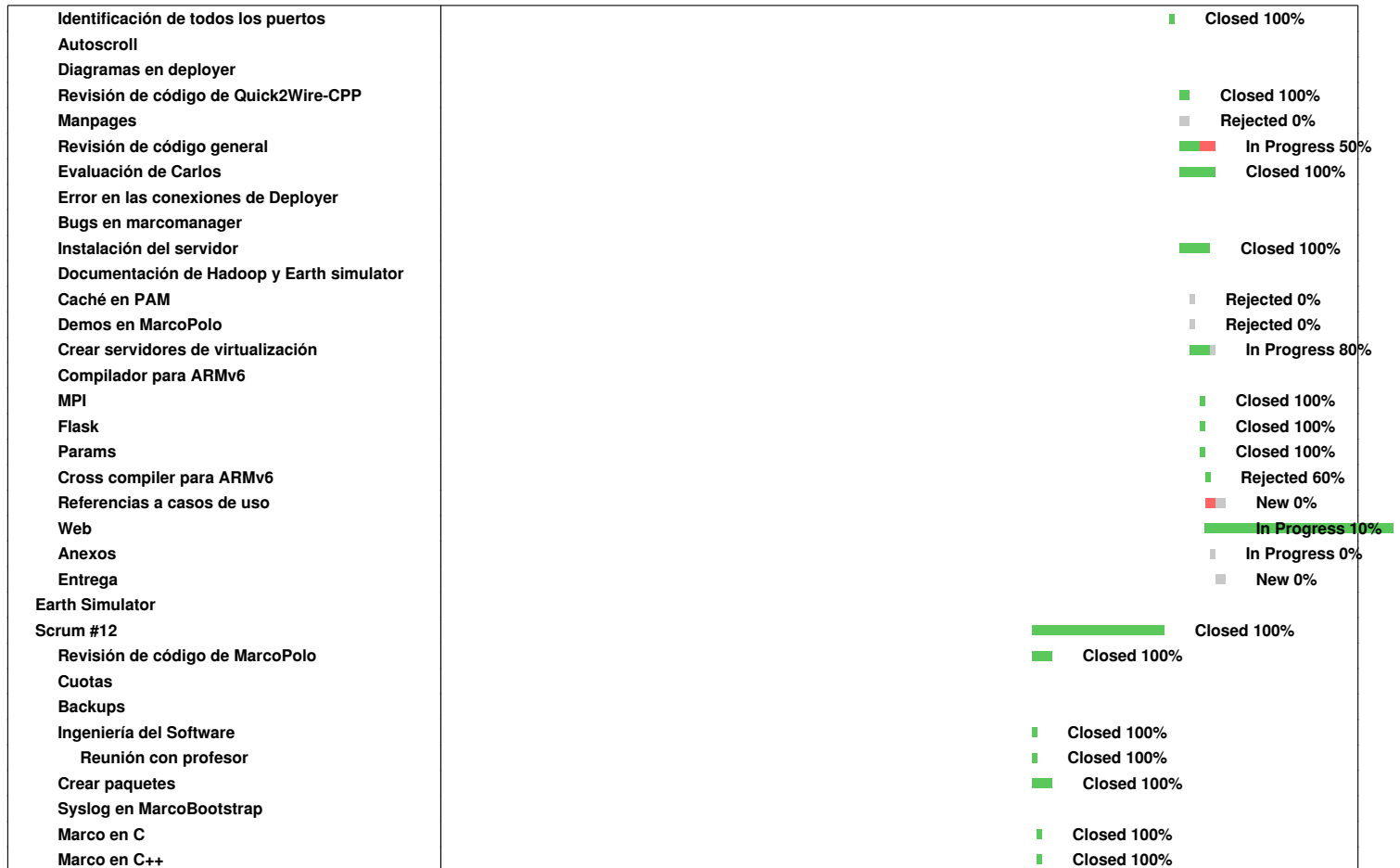


Diagrama de Gannt (parte 6)

Marco en Java	<div><div></div></div> Closed 100%
Hadoop	<div><div></div></div> Rejected 0%
C++ binding	
Documentación general	
Documentación general	
Scrum #13	<div><div></div></div> Closed 100%
Empaquetar deployer	<div><div></div></div> Closed 100%
Polo con SSL y token de autenticación	<div><div></div></div> Closed 100%
Evaluaciones de usuario	<div><div></div></div> Closed 80%
Reunión con los tutores	<div><div></div></div> Closed 100%
Instalación de MarcoPolo en las aulas	<div><div></div></div> Closed 100%
Entrega	<div><div></div></div> Closed 100%
Empaquetar marcomanager	<div><div></div></div> Closed 100%
Error de codificación en MarcoPolo	<div><div></div></div> Closed 100%
Request_for("RFC")	<div><div></div></div> Closed 100%

## Gestión de la calidad

La gestión de la calidad del código se realiza mediante la combinación de las siguientes estrategias:

- **Pruebas automatizadas:** Utilizando herramientas como unittest<sup>4</sup> o Trial<sup>5</sup> se ha realizado un desarrollo dirigido por pruebas.
- **Pruebas manuales:** De forma similar a las pruebas automatizadas varios paquetes *software* han sido creados utilizando pruebas manuales.
- **Evaluaciones de usuario:** En aquellos casos en los que el *software* creado deba ser utilizado de forma directa por un usuario humano se evaluará esta interacción mediante procedimientos de evaluación.
- **Seguimiento de guías de estilo de código y utilización de herramientas de verificación:** Guías como la PEP 0008<sup>6</sup> o Pylint<sup>7</sup> asisten al desarrollador a la hora de crear código fuente que siga las convenciones propias de cada lenguaje. El uso de las mismas permite que el código sea más fácil de comprender por otros desarrolladores. 2to3<sup>8</sup> y 3to2<sup>9</sup>, entre otras utilidades, verifican la compatibilidad de código Python entre versiones. La guía de portabilidad de Python<sup>10</sup> ha sido de gran utilidad en esta tarea.

## Comunicación

La comunicación entre los diferentes integrantes del equipo se basa en reuniones presenciales entre todos los miembros. No ha sido necesaria una mayor formalización debido al reducido tamaño del equipo.

## Análisis de riesgos

Uno de los mayores riesgos que las características del proyecto define es el incumplimiento de las estrictas fechas de desarrollo. Emplear demasiados recursos y tiempo en una tarea que no satisfaga los requisitos marcados inicialmente dañaría severamente el proyecto. Sin embargo, debido a la gran adaptabilidad que la metodología utilizada aporta ha sido posible determinar de forma temprana los problemas de una propuesta de solución.

## Control de versiones

Es importante mantener un control de las diferentes versiones en desarrollo y productos creados durante el proceso de desarrollo. Dicho control facilita la gestión del proyecto, permite retornar el proyecto a un estado anterior o mantener de forma concurrente varias ramas de desarrollo.

Dado que el proyecto implica la construcción de un conjunto de herramientas que compongan un sistema, se utilizará un repositorio de código para cada una de ellas. El gestor elegido es **git**<sup>11</sup>, debido a su carácter distribuido, posibilidad de trabajar con varias ramas de desarrollo y experiencia previa. Todo el código se almacena en el proveedor de alojamiento **Bitbucket**<sup>12</sup>, actuando como copia de seguridad de los datos recogidos en el equipo local.

<sup>4</sup><https://docs.python.org/2/library/unittest.html>

<sup>5</sup><http://twistedmatrix.com/trac/wiki/TwistedTrial>

<sup>6</sup>Python Enhancement Proposal 0008: <https://www.python.org/dev/peps/pep-0008/>

<sup>7</sup><http://www.pylint.org/>

<sup>8</sup><https://docs.python.org/2/library/2to3.html>

<sup>9</sup><https://pypi.python.org/pypi/3to2/1.1.1>

<sup>10</sup><https://docs.python.org/3/howto/pyporting.html>

<sup>11</sup><http://git-scm.com/>

<sup>12</sup><https://bitbucket.org/>

Todos los repositorios son públicos, y están disponibles en el DVD adjunto a la documentación física y en la url <https://bitbucket.org/Alternhuman>.

Como conclusión cabe valorar el éxito de la metodología utilizada que ha facilitado, si no posibilitado, el buen devenir de este proyecto.