

Cazasteroides

Plan de Trabajo
22/02/2017

Realizado por Miguel Núñez Díaz-Montes
Tutelado por Francisco Rosales García
Febrero-Junio del 2017

Índice

Descripción General.....1

Lista de Tareas.....2

Diagrama de Gantt.....3

Propuesta de Trabajo Original.....4

Descripción General

La idea de este proyecto es acercar la astronomía a la ciudadanía, convirtiéndoles en cazadores de asteroides. Los usuarios tendrán acceso a recortes de fotografías reales obtenidas a través de telescopios. En estas imágenes se buscarán los asteroides y los hallazgos se enviarán al sistema donde el resto de usuarios podrán dar su opinión. Para aumentar la popularidad y conseguir una gran comunidad activa de usuarios, se va a gamificar el sistema.

Gamificar consiste en acercar el concepto de juego a una idea. En este caso sería atribuir puntos a hallazgos y votaciones y así aumentar el karma, proponer retos a tus amigos, dar recompensas a los mejores jugadores...

Considero importante añadir que este trabajo tiene parte ya avanzada, aunque no está correctamente documentado y es inestable. Esto conlleva que parte del proyecto se centre en documentar y estabilizar el sistema. Además, van a trabajar otros dos desarrolladores por lo que la comunicación y el buen reparto de tareas va a tener gran importancia.

El sistema actual se puede dividir en las siguientes partes:

- **Cliente:** programado en HTML5, CSS y JavaScript, concretamente utilizando el *framework* AngularJS. Partiendo de este código, utilizando la herramienta Apache Cordova se construyen aplicaciones nativas para Android e iOS.
- **Servidor:** tendríamos un servidor programado en Node.js.
- **Base de datos:** concretamente MongoDB, tipo de base de datos que se caracteriza por tener un esquema dinámico de datos, aunque utilizaremos Mongoose, un plugin de Node.js, que a parte de realizar las consultas, permite modelar los datos.
- **Procesador de imágenes:** es el encargado de realizar todo el procesamiento de las imágenes de los telescopios, como es el recorte de la imagen original, la escala, y la compresión. Además, genera la animación de la observación.

Este proyecto se va a centrar principalmente en el cliente y en el servidor, los cuales, para comunicarse siguen una arquitectura REST.

El código del proyecto se encuentra alojado en BitBucket, una implementación de Git en la nube. Con esto conseguimos un control de versiones que tiene interesantes funcionalidades para el trabajo en grupo, como pueden ser:

- **Control de versiones:** permite tener controlados todos los cambios, si fuese necesario se podría ver un estado del proyecto pasado, o volver a el.
- **Issues:** permite crear incidencias con errores encontrados o funcionalidades que sean necesarias añadir.
- **Branches:** permite trabajar en paralelo sobre dos funcionalidades distintas sin que un desarrollo afecte a otro.

Toda la documentación va a estar alojada en una carpeta compartida en Google Drive que permitirá a los supervisores del proyecto llevar un control regular del trabajo, además de que permite realizar comentarios y ofrece un sencillo control de versiones.

Como metodología se va a seguir un desarrollo ágil. Como ahora mismo no se sigue una metodología de este tipo se va a llevar a cabo una transición. Destacando las siguientes métodos a seguir:

- **Reuniones** semanales para ver cual ha sido el avance.
- **Ramas:** se trabajará en distintas ramas lo que permitirá trabajar en distintos elementos sin que unos alteren al resto.
- **Sprints:** Se marcarán objetivos a corto plazo de los requisitos más importantes intentando solo trabajar en una funcionalidad a la vez. Esto lo realizaremos mediante *features* (bifurcaciones de la rama principal). Cuando se termine una funcionalidad esta se unirá a la rama de desarrollo y se cerrará (si fuera necesario).

Lista de Tareas

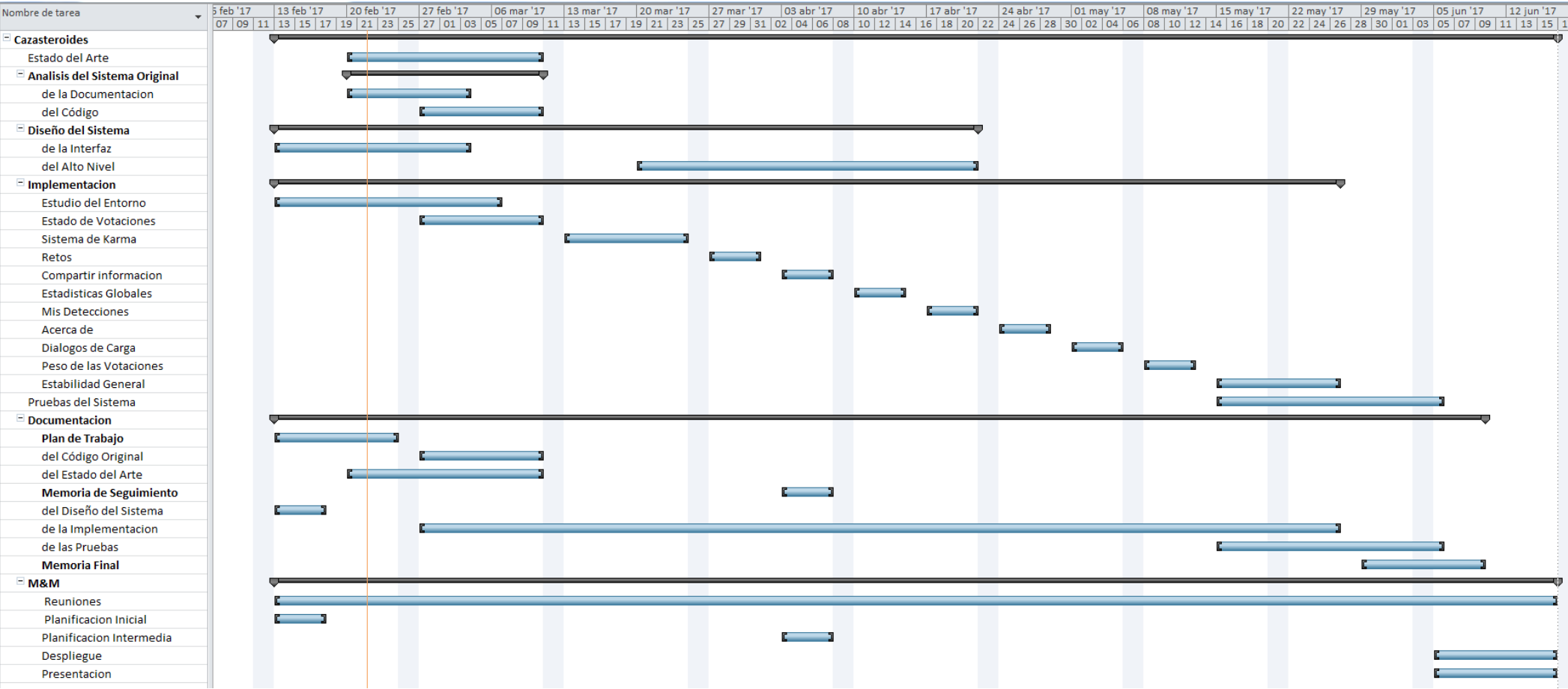
Para definir estas tareas se han utilizado las ideas originales propuestas por el tutor, de los desarrolladores originales y mis aportaciones según mis conocimientos.

Se han dividido estas tareas en distintos subgrupos para su mejor clasificación. Esta lista de tareas se puede apreciar en la siguiente tabla.

Grupos	Tareas		
Estado del Arte	Análisis del Estado del Arte		
Análisis del Sistema Original	de la Documentación	del Código	
Diseño del Sistema	de la Interfaz	de Alto Nivel	
Implementación	Estudio del Entorno	Estado Votaciones	Sistema de Karma
	Retos	Compartir Información	Estadísticas Globales
	Mis Detecciones	Acerca De	Diálogos de Carga
	Peso de las votaciones	Estabilidad General	
Pruebas	Pruebas del Sistema		
Documentación	Plan de Trabajo	del Código Original	del Análisis del Sistema
	del Estado del Arte	Memoria de Seguimiento	del Diseño del Sistema
	de la Implementación	de las Pruebas	Memoria Final
M&M	Reuniones	Planificación Inicial	Presentación
	Planificación Intermedia	Despliegue	

Tras concretar estas tareas. Se han repartido durante todas las semanas teniendo en cuenta las semanas destinadas al trabajo. Se ha empezado a realizar este trabajo el día 13 de febrero, correspondiendo a la semana 1, y se ha planificado su finalización el día 19 de junio, correspondiendo a la semana 20.

Diagrama de Gantt



Propuesta de Trabajo Original

Título del trabajo	Cazasteroides
Resumen general del trabajo	Cazasteroides es un proyecto de ciencia ciudadana que consiste en la búsqueda y localización por parte de los usuarios de asteroides cercanos a la tierra, denominados NEAs (del inglés <i>Near Earth Asteroids</i>) o NEOs (<i>Near Earth Objects</i>). El alumno deberá gamificar la aplicación ya existente para de este modo incrementar la participación de los usuarios. Para ello hará uso de distintas técnicas de gamificación como sistemas de recompensas y desafíos entre jugadores. Por ejemplo, ranking de usuarios más rápidos, etc. Asimismo se permitirá a los usuarios publicar sus logros en redes sociales como Twitter y Facebook e invitar a nuevos usuarios.
Lista de objetivos concretos	<ul style="list-style-type: none"> • Familiarizarse / documentarse sobre las técnicas de gamificación • Gamificar la aplicación Cazasteroides • Mostrar los resultados en tiempo real
Desglose de la dedicación total del trabajo en horas	<ul style="list-style-type: none"> • Estado del arte de juegos aplicados a ciencia ciudadana (20 horas) • Análisis de la aplicación Cazasteroides (20 horas) • Diseño de la aplicación (70 horas) • Implementación de la aplicación (120 horas) • Despliegue y pruebas (44 horas) • Documentación (50 horas)
Conocimientos recomendados para hacer el trabajo	<ul style="list-style-type: none"> • Tecnologías Web: CSS3, HTML5 y JavaScript. • Desarrollo de aplicaciones móviles responsive multiplataforma, concretamente en Apache Cordova.