****

Desarrollo de Aplicaciones Web Curso 2015-2016

Aplicación para lanzar pruebas unitarias en JavaScript de forma remota

****

DOP TESTER

4/5/2016

David Villaluenga Casasola

**CICLO FORMATIVO DE GRADO SUPERIOR**

**Desarrollo de aplicaciones Web en entorno Internet**

**AUTOR**

**David Villaluenga Casasola**

**TUTOR**

**José María Alonso Josa**



**Contenido**

[1. Justificación del proyecto y objetivos 3](#_Toc447135932)

[2. Introducción 4](#_Toc447135933)

[3. Metodología y desarrollo del proyecto 4](#_Toc447135934)

[4. Resultados y discusión 4](#_Toc447135935)

[5. Conclusiones 4](#_Toc447135936)

[6. Bibliografía y referencias 4](#_Toc447135937)

[7. Anexos/ Otros 4](#_Toc447135938)

# **Justificación del proyecto y objetivos**

Con la realización de este proyecto, se pretende realizar una aplicación que permita la automatización de las pruebas unitarias sobre un software escrito en lenguaje JavaScript. Este sistema de automatización de pruebas unitarias, facilitará la realización de proyectos siguiendo un desarrollo guiado por pruebas TDD (Test-Driven Development) o BDD ((Behaviour Driven Development), permitiendo así el desarrollo de software con una calidad superior.

La elección de este proyecto se debe, a la oportunidad de mejorar en el aprendizaje sobre la metodología TDD y BDD, tecnología muy interesante desde el punto del desarrollo de software, pues supone una metodología diferente a la estudiada en el curso. Además supone una forma de desarrollo muy adecuada para la programación de software comercial, pues es una forma fácil y comprensible de llegar a un acuerdo entre el desarrollador de software y el cliente, dado que fija unos objetivos reales a cumplir.

La metodología de desarrollo orientado a pruebas está muy ligada al desarrollo ágil, por lo tanto este tipo de desarrollo será el más adecuada a la hora de proporcionar una metodología de programación en el proyecto.

La realización de este proyecto permitirá además adquirir conocimientos de otros modelos de programación en el lado del servidor, como pueden ser el *software stack* MEAN. Así como el sistema de repositorios de GitHub.

Durante la realización de este proyecto se realizará una aplicación Web, que permita registrarse con una cuenta de un repositorio de software, que automatice la ejecución de los test sobre el código JavaScript dentro de estos repositorios, así como ver los resultados de los test realizados.

Los test se ejecutarán de forma autónoma cada vez que el repositorio sufra al algún cambio, permitiendo así al desarrollador conocer rápidamente el alcance de los cambios realizados en el código.

# **Introducción**

DOP Tester, será una aplicación que con muy pocos pasos, permita a sus usuarios automatizar el proceso de realización de pruebas sobre su código JavaScript, sin necesidad de preocuparse de realizarlas cada vez que el software sufra algún cambio.

Esta interactividad entre el cambio en el código y la respuesta inmediata de los resultados de los test, permitirá conocer de una forma rápida si los cambios realizados en el software son correctos, o en cambio se ha cometido algún error en su codificación, permitiendo de esta forma poder subsanar los errores en el momento en que se han producido o si esto no es posible volver a una versión anterior del software, sin necesidad de una regresión mayor que un ciclo de codificación.

Si esta operatividad rápida le unimos una interfaz Web sencilla, y la posibilidad de gestión de los repositorios en la misma interfaz. Obtendremos una aplicación Web sencilla que facilite la aplicación de una metodología de desarrollo ágil, metodología en auge en la actualidad.

# **Metodología y desarrollo del proyecto**

## **Metodologia ágil Scrum**

La metodología empleada en el desarrollo de la aplicación será la de desarrollo ágil, más especificamiente la metodología Scrum. Aunque está tecnología esta pensado para proyectos con un cliente y con un equipo de trabajo, adaptaré el proceso a la situación del proyecto actual, es decir, el cliente y el equipo de trabajo seré yo.

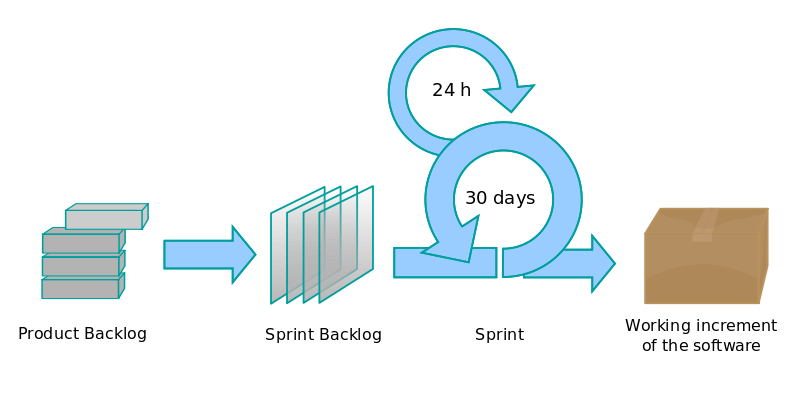


Ilustración 1 Ciclo de desarrollo Scrum

El procedimiento de desarrollo Scrum, se caracteriza por una estratégia de desarrollo por iteraciones o programación incremental, cada iteración proporcionará un software funcional que cumpla en parte con las especificaciones dictadas o redactadas por el cliente. A su vez cada iteración esta compuesta por reuniones diarias del equipo de trabajo, para conocer el estado del desarrollo así como si ha surgido algun problema o incidencia, que deba ser resuelto como una tarea nueva.

Cada iteración consta de un conjunto de *Sprints*, estos son los periodos de tiempo durante los cuales se realiza el trabajo, en este proyecto cada sprint tendrá una duración de dos semanas, haciendo coincidir su finalización con las reuniones con el tutor, depués de cada sprint se mostrará el trabajo realizado que debe constituir un producto entregable en si mismo.

Para la realización de la planificación del proyecto se realizan tres documentos:

* Product backlog 🡪 Es un documento de alto nivel donde constan los requerimientos que el proyecto debe cumplir al finalizar el mismo, este documento puede ser actualizado en cualquier momento con nuevas especificaciones.
* Sprint backlog 🡪 Es un subconjunto del product backlog donde constan los requisitos que serán desarrollados durante el siguiente sprint. Los requisitos en este documento se subdividiran en tareas las cuales deben tener, siempre que sea posible, una duración menor de 16 horas.
* Burn down chart 🡪 Gráfico, que muestra la cantidad de requisitos del product backlog pendientes al final de cada sprint. La gráfica muestra en el eje Y las horas necesarias para concluir el proyecto y en el eje X los sprint realizados, mediante este gráfico es posible conocer como se esta desarrollando el proyecto, en cuanto a plazos se refiere.

**Metodología**

*Se realiza una aproximación del desarrollo, o duración del mismo pero al tratarse de un desarrollo ágil esta aproximación cuanto mas alejada este del inicio, mas margen de error tendrá., además estos sprint podrian quedar modificados dependiendo del desarrollo de los sprints anteriores.*

**Product backlog**

1. Realizar una aplicación Web.
2. Alta de los usuarios con su cuenta de GitHub.
3. Enlazar sus repositorios de GitHub con la aplicación.
4. Notificación de los resultados de los test.
5. Mostrar las estadísticas de los test realizados.

**Product Sprint**

* Sprint 0 (4/4/2016 - 12/4/2016) [19 horas]

Diseño de las vistas 🡪 3 horas.

Diseño de los modelos y bases de datos 🡪 2 horas.

Estudio de la tecnología MEAN 🡪 14 horas.

* Sprint 1 (13/4/2016 - 27/4/2016) [36 horas]

Estudio del sistema de notificación 🡪 4 horas.

Implementación del servidor Web 🡪 3 horas.

Implementación de las vistas 🡪 12 horas.

Implementación de los modelos y bases de datos 🡪 2 horas.

Estudio del API de GitHub 🡪 10 horas.

Implementación del alta de usuarios 🡪 5 horas.

* Sprint 2 (28/4/2016 - 11/4/2016) [33 horas]

Registro de los usuarios autenticados 🡪 4 horas.

Implementación del sistema de notificación 🡪 12 horas.

Acceso a los repositorios de los usuarios 🡪 5 horas.

Implementación del lanzamiento de pruebas 🡪 12 horas.

* Sprint 3 (12/4/2016 - 25/4/2016) [32 horas]

Implementación de la recogida de los resultados de las pruebas 🡪 16 horas.

Realización del dossier del proyecto 🡪 16 horas.

* Sprint 4 (25/4/2016 – 30/4/2016) [8 horas]

Pruebas de la aplicación 🡪 6 horas.

Maquetado del proyecto 🡪 2 horas.

**Burn down chart**

# **Resultados y discusión**

# **Conclusiones**

# **Bibliografía y referencias**

<http://agilemanifesto.org/> 🡪 Manifesto for Agile Software Development

<http://www.martinalaimo.com/es/> 🡪 Blogger de desarrollo ágil con Scrum.

<https://es.wikipedia.org/>

<https://en.wikipedia.org/>

# **Anexos/ Otros**