

Evolución y Gestión de la Configuración

Agor@Us

Autenticación Mejoras

Grupo 1 – ID de Opera 7

Fidel Mazo Delgado
Alejandro Garrido Resina
Alejandro Guerrero Montoro
Alejandro Román Rodríguez
Julio Márquez Castro
María Remedios Dans Caballero

Contenido

1.	Respuestas a comentarios del profesor.....	1
1.1.	Mejorar la defensa. Llegan demasiado justos a la defensa...	1
1.2.	Nula o casi nula integración con el resto de subsistemas.....	1
1.3.	El despliegue no parece coherente con lo que está en local y con las ramas.....	1
1.4.	Hay scripts para Jenkins, pero en local sin usar la infraestructura común.....	1
1.5.	Parecen no tener claro en la practica el proceso de desarrollo, especialmente en la gestión del código. La documentación no aporta mucho más.....	1
1.6.	La documentación debe mejorar considerablemente.....	2
1.7.	No entrega diario de grupo	2
1.8.	Es difícil llegar al repositorio.....	2
2	Otras mejoras.....	2
2.1.	Travis.....	2
2.2.	ReCaptcha.....	3
2.3.	Api.....	3

1. Respuestas a comentarios del profesor

1.1. Mejorar la defensa. Llegan demasiado justos a la defensa

Por un fallo en la planificación del grupo no llegamos bien preparados para la entrega, pero tras varios días de reunión, y trabajo en grupo nos sentimos más preparados y con los conocimientos requeridos en la asignatura más asentados.

Hemos mejorado la forma de trabajar en grupo y usar herramientas como GitHub, Travis, Jenkins y algunos otros conocimientos como jerarquías, procedimientos y dinámicas de grupo.

1.2. Nula o casi nula integración con el resto de subsistemas

Tras la revisión de varios proyectos de otros subsistemas, encontramos varios subsistemas que lo hacen como por ejemplo Deliberations, que consume de nuestra api para loguearse en su aplicación, como explicamos más concretamente en el documento principal.

1.3. El despliegue no parece coherente con lo que está en local y con las ramas

Como hemos mencionado anteriormente, tenemos mayor control sobre GitHub y todas sus ventajas. Gestionamos mucho mejor los cambios en las diferentes ramas y las fusiones entre ellas, y la actualización a la rama master, rama principal de nuestro sistema y usada por ejemplo para la integración continua con Jenkins.

1.4. Hay scripts para Jenkins, pero en local sin usar la infraestructura común

Probamos a usar Jenkins en local para ejecutar los test y desplegar el sistema en nuestro servidor aislado, pero tras algunos problemas decidimos usar Travis para los test y desplegar en nuestro servidor aislado de forma manual. El despliegue al Jenkins de infraestructura común lo teníamos implementado, pero nos faltaba un poco de soltura en su uso, cosa que ya hemos mejorado.

1.5. Parecen no tener claro en la practica el proceso de desarrollo, especialmente en la gestión del código. La documentación no aporta mucho más

Como mencionamos anteriormente, creemos haber mejorado considerablemente en este aspecto, con un correcto uso de GitHub y una mayor sincronización entre los integrantes del grupo.

1.6. La documentación debe mejorar considerablemente

Tras revisar exhaustivamente la wiki de la asignatura y toda la documentación requerida, hemos trabajado mayoritariamente en equipo para un buen desarrollo de la documentación. Este ha sido uno de los puntos que más horas nos ha “robado” en la realización de la mejora.

1.7. No entrega diario de grupo

El grupo tenía realizado un control de horas antes de la entrega anterior, pero no estaba actualizado y se decidió no entregarlo. Para la mejora hemos realizado un completo y actualizado diario.

1.8. Es difícil llegar al repositorio

En la nueva documentación hemos puesto diferentes enlaces para acceder a toda la documentación, añadiendo un valor añadido a la información del proyecto facilitando su comprensión.

2. Otras mejoras

2.1. Travis

Introducimos la herramienta Travis (<https://travis-ci.org/>) en nuestro sistema para ampliar la integración continua. Esta herramienta facilita la ejecución de los test de nuestra aplicación. Los ejecuta automáticamente una vez se suba algún cambio a alguna de las ramas de GitHub.

Explicamos más sobre el procedimiento que realizamos en el documento principal “Autenticación - Grupo 7 - Documento Final” en el punto 7 “Gestión de la integración continua.”

2.2. ReCaptcha

Estudiamos y revisamos la configuración de seguridad por Captcha en el registro de nuevos usuarios de nuestro sistema. Esta herramienta ya fue implementada el año pasado, pero era necesario el registro de nuestro nuevo dominio en la api de Google sobre Recaptcha (<https://www.google.com/recaptcha/intro/comingsoon/invisiblebeta.html>).

Una vez reconfigurado el proyecto esta herramienta funciona correctamente en nuestro proyecto.



The image shows a registration form on a magenta background. It includes three input fields: 'Correo electrónico:' with an envelope icon, 'Contraseña:' with a lock icon, and 'Comunidad autónoma:' with a location pin icon. Below these is a ReCaptcha widget with the text 'No soy un robot' and a checkbox. To the right of the checkbox is the ReCaptcha logo and the text 'reCAPTCHA Privacidad - Condiciones'. At the bottom center is a blue button labeled 'Enviar'.

2.3. Api

Realizadas algunas mejoras en cuanto la api que teníamos.

Cambios:

- Añadido soporte en la API REST para acceder mediante URLs amigables
- Cambiar método de devolver usuario para no dar contraseña.
- Mejorado control de errores, ahora se muestran como código json según las buenas practicas, a través de un método que muestra el código de error, un mensaje y los parámetros que hayan sido especificados.
- Mostrar error si has puesto un login que no es correcto, ahora muestra una ventana.