

Cómo hacer que tus tests te mantengan enfocado

Al comenzar la charla, destacaré la importancia de tener una *test suite* rápida apoyándome en lo que nos dice la psicología al respecto, especialmente en cómo afectan a nuestro cerebro las distintas escalas de tiempo para obtener el resultado de algo que hemos hecho (*feedback*), siendo un *feedback-loop* corto el adecuado para seguir en el *mood* y un *feedback-loop* largo contraproducente para nuestro enfoque.

La charla se centrará en qué malas prácticas debemos evitar, y posteriormente en cómo obtener una *test suite* sana, haciendo especial hincapié en los tests de aceptación. Para ello, introduciré la arquitectura "Puertos y Adaptadores" y enseñaré cómo aprovecharla para sacarle el máximo partido para nuestra *test suite*.

Detalles de la propuesta:

- Tipo de propuesta: Charla larga / Presencial
- Descripción:

Todos lo hemos visto, parece que a la hora de hablar de diseño de software todo se centra en la parte de código de producción. Nombres de métodos que revelan intención, clases cohesivas, poco acoplamiento... pero poco se habla de la aplicación de buenas prácticas y principios en nuestra otra "gran parte del proyecto": nuestra *test suite*.

Parece que por alguna razón, cuando de hablar de tests se trata, relajamos todas esas buenas prácticas, principios y búsqueda continua de que todo funcione de forma óptima, algo que con el paso del tiempo, conduce hacia tests de baja calidad (*slow feedback-loop*, *flakiness* y *fragility*).

A diferencia de código de producción, ningún *Product Owner* se quejará de que los tests van lentos, y tampoco saltará una alerta en slack para indicarnos de que nuestros tests van más lentos de lo debido.

Es a los propios desarrolladores a quienes más influye una *test suite* en mal estado, siendo la causa principal un *feedback-loop* lento, que hace que nuestro enfoque sobre lo importante empeore, afectando así en gran medida a nuestra *overall Developer Experience* por ser algo tan importante de nuestro día a día.

Por todo esto podemos decir que tener una *test suite* sana es de vital importancia para cualquier proyecto, por ello, en esta charla quiero mostrar:

- Importancia de tener una *test suite* rápida desde el punto de vista de la psicología
- Arquitectura Puertos y Adaptadores y sus beneficios *out of the box* para testing
- Diferencias entre Test de Aceptación integrado y Test de Aceptación desacoplado
- *Contract Testing* / *Adapter Tests* y cómo pueden ayudarnos para nuestros Tests de aceptación
- Herramienta de análisis estático de código para validar nuestra arquitectura, en caso de que de tiempo (php architecture tester / phpat)

¡Incluiré ejemplos de código también!

- Público objetivo:

Cualquier interesado en mantener una *test suite* sana. Developers, QAs, DevOps

Ponente:

- Nombre: Filis Futsarov
- Bio:

Passionate software developer for building high-quality and human-centred tech. Very interested in testing as a living documentation tool for developers.

Info personal:

- Web personal: <https://filis.me>
- Twitter: <https://twitter.com/filisdev>
- GitLab (o cualquier sitio de código colaborativo) o portfolio: <https://github.com/filisko>

Condiciones aceptadas

- Acepto seguir el código de conducta (<https://eslib.re/conducta>) durante mi participación en el congreso

[Volver al índice](#)

© 2022 **esLibre**

Sitio web generado con [Jekyll](#) publicado bajo una licencia [Creative Commons Atribución-CompartirIgual 4.0 Internacional \(CC BY-SA 4.0\)](#).

