**2. Descripción de las tecnologías**

A continuación, vamos a analizar brevemente cada una de las tecnologías seleccionadas en el trabajo anterior, PHPUnit y Codeception.

**2.1 Descripción de la tecnología PHPUnit**

PHPUnit es un framework open source para el desarrollo, orientado a pruebas o Test-driven development (TDD) para cualquier código PHP. Es decir, es un framework que nos ayuda a probar nuestro código creando pequeñas unidades que revisan funcionalidades puntuales del código y probando que funcionan como deben, además de la posibilidad de automatizar estas pruebas para ejecutarlas frecuentemente, tanto como el código cambie.



Figura 1

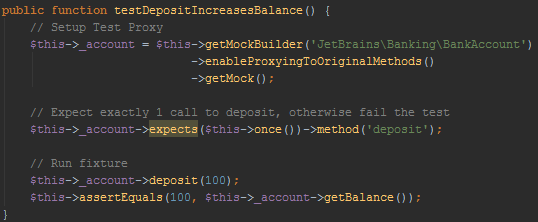


Figura 2

URL1,2: [https://blog.jetbrains.com/phpstorm/2014/05/phpunit-4-in-phpstorm-8-eap/](https://blog.jetbrains.com/phpstorm/2014/05/phpunit-4-in-phpstorm-8-eap/" \t "_blank)

**3.4 Categoría D: Velocidad de ejecución de prueba**

3.4.1 Criterio D.1: Velocidad pruebas unitarias

Nombre del criterio: Velocidad de la ejecución al realizar las pruebas unitarias

Descripción: Velocidad que se demora el entorno de testeo automatizado para PHP en ejecutar las pruebas unitarias. Este valor se determina en función del tiempo que tarda el resto de software similares a este en realizar el mismo test unitario, considerándose como valores altos los más rápidos con mínimas diferencias y lentos los que más tiempo necesitan.

Tipo de valor: Alto / Medio / Bajo.

3.4.2 Criterio D.2: Velocidad pruebas funcionales

Nombre del criterio: Velocidad de la ejecución al realizar las pruebas funcionales

Descripción: Velocidad que se demora el entorno de testeo automatizado para PHP en ejecutar las pruebas funcionales. Este valor se determina en función del tiempo que tarda el resto de software similares a este en realizar el mismo test funcional, considerándose como valores altos los más rápidos con mínimas diferencias y lentos los que más tiempo necesitan.

Tipo de valor: Alto / Medio / Bajo.

3.4.3 Criterio D.3: Velocidad pruebas de aceptación

Nombre del criterio: Velocidad de la ejecución al realizar las pruebas de aceptación

Descripción: Velocidad que se demora el entorno de testeo automatizado para PHP en ejecutar las pruebas de aceptación. Este valor se determina en función del tiempo que tarda el resto de software similares a este en realizar el mismo test de aceptación, considerándose como valores altos los más rápidos con mínimas diferencias y lentos los que más tiempo necesitan.

Tipo de valor: Alto / Medio / Bajo.

3.4.4 Criterio D.4: Velocidad pruebas de integración

Nombre del criterio: Velocidad de la ejecución al realizar las pruebas de integración

Descripción: Velocidad que se demora el entorno de testeo automatizado para PHP en ejecutar las pruebas de integración. Este valor se determina en función del tiempo que tarda el resto de software similares a este en realizar el mismo test unitario, considerándose como valores altos los más rápidos con mínimas diferencias y lentos los que más tiempo necesitan.

Tipo de valor: Alto / Medio / Bajo.

3.4.5 Criterio D.5: Velocidad pruebas de regresión

Nombre del criterio: Velocidad de la ejecución al realizar las pruebas de regresión

Descripción: Velocidad que se demora el entorno de testeo automatizado para PHP en ejecutar las pruebas de regresión. Este valor se determina en función del tiempo que tarda el resto de software similares a este en realizar el mismo test unitario, considerándose como valores altos los más rápidos con mínimas diferencias y lentos los que más tiempo necesitan.

Tipo de valor: Alto / Medio / Bajo.

3.4.6 Criterio D.6: Velocidad pruebas de estrés

Nombre del criterio: Velocidad de la ejecución al realizar las pruebas de estrés

Descripción: Velocidad que se demora el entorno de testeo automatizado para PHP en ejecutar las pruebas de estrés. Este valor se determina en función del tiempo que tarda el resto de software similares a este en realizar el mismo test unitario, considerándose como valores altos los más rápidos con mínimas diferencias y lentos los que más tiempo necesitan.

Tipo de valor: Alto / Medio / Bajo.

3.4.7 Criterio D.7: Velocidad pruebas de rendimiento

Nombre del criterio: Velocidad de la ejecución al realizar las pruebas de rendimiento

Descripción: Velocidad que se demora el entorno de testeo automatizado para PHP en ejecutar las pruebas de rendimiento. Este valor se determina en función del tiempo que tarda el resto de software similares a este en realizar el mismo test unitario, considerándose como valores altos los más rápidos con mínimas diferencias y lentos los que más tiempo necesitan.

Tipo de valor: Alto / Medio / Bajo.

**5. Comparación de las tecnologías**

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERIOS** | **PHPUnit** | **Codeception** | **COMENTARIOS** |
| A.1 | Gratis | Gratis | Se pueden hacer donativos. |
| A.2 | Inglés | Inglés |  |
| A.3 | Alta | Alta |  |
| A.4 | BSD | Open Source | Ambas licencias están relacionadas con el software libre. En el caso de Codeception es MIT license. |
| A.5 | Alta | Media | En el caso de codeception se limita a prácticamente a la documentación oficial. |
| B.1 | Baja | Alta | Codeception requiere una configuración inicial más compleja. |
| B.2 | Si | No | La librería Infection estará próximamente disponible en Codeception. |
| B.3 | Si | Si |  |
| B.4 | Si | Si |  |
| B.5 | Baja | Alta | Un análisis con Codeception puede tardar algo más de 1 min, mientras que en PHPunit estaría en torno a 15 min. |
| B.6 | Media | Baja | Los resultados con codeception son fácilmente interpretables por usuarios sin conocimientos técnicos. |
| C.1 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.2 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.3 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.4 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.5 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.6 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.7 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| D.1 | Alta | Alta | Al ser pruebas unitarias (generales) se realizan rápidamente, no se obtienen diferencias notables. |
| D.2 | Alta | Alta | Del mismo modo que en el caso anterior, ambas pruebas funcionales se realizan rápidamente, no encontramos diferencias destacables como para decantarnos por una tecnología u otra por esta característica. |
| D.3 | Baja | Media | En este caso si tenemos una pequeña diferencia de rapidez en las pruebas de aceptación. Son más rápidas en el caso de Codeception. |
| D.4 | Baja | Media | Las pruebas de integración combinan los distintos módulos de una aplicación. Esto es una tarea compleja que depende en cierta medida de como estén codificadas las pruebas. Además, también influye como de optimizadas estén las librerías que use el entorno de pruebas y por lo tanto la velocidad de las mismas está influida por el tipo de entorno que se esté usando. En este caso Codeception usa librerías más optimizadas. |
| D.5 | Baja | Baja | La velocidad de las pruebas de regresión está fuertemente relacionada con lo optimizado que se encuentre el entorno de trabajo que se esté utilizando. Las pruebas de regresión consisten en probar completamente una versión nueva de una aplicación y por lo tanto suele ser un proceso lento. |
| D.6 | Media | Media | Las pruebas de estrés consisten en llevar a condiciones extremas a la aplicación. El gran volumen de datos hace que esta tarea conlleve un tiempo de ejecución medio. Ambos entornos de trabajo están preparados para optimizar el tiempo en este tipo de pruebas. |
| D.7 | Alta | Alta | Las pruebas de rendimiento están relacionadas con las pruebas de estrés. La diferencia es que no suelen llevar a condiciones extremas a la aplicación, sino que se estudia el tiempo de respuesta de la misma. En este sentido, ambos entornos están muy optimizados en este tipo de pruebas. |

**6. Recomendaciones**

Basándonos en los criterios de valoración de cada tecnología planteamos las siguientes posibles situaciones:

**6.1 Situación 1**

6.1.1 Descripción de la situación

La empresa Knuto desarrolla aplicaciones de Gestión Medioambiental en su propio departamento de desarrollo. La complejidad de estas aplicaciones es bastante alta, incluso incorporan tecnología GIS. La empresa desea conocer si los programas desarrollados son óptimos y fiables para poder ser lanzados posteriormente al mercado. Los resultados obtenidos los requieren también desde el departamento legal para justificar la adecuación de la aplicación a la normativa establecida. El equipo que se va a encargar de la automatización del testing tiene amplios conocimientos y experiencia. Para ello, plantean utilizar una de las dos herramientas que ha propuesto el jefe de departamento de desarrollo, que son Codeception y PHPUnit.

6.1.2 Recomendación de tecnología a utilizar

En esta situación el departamento se decanta por elegir Codeception, las ventajas que se han encontrado y por las que se decantan por esta tecnología las mostramos a continuación en una tabla:

|  |  |  |
| --- | --- | --- |
| **Criterios relevantes para la decisión** | **Ventajas PHPUnit** | **Ventajas Codeception** |
| Velocidad de ejecución |  | Al tratarse de aplicaciones de una gran cantidad de líneas, Codeception tarda menos tiempo en realizarlas. |
| Dificultad para la puesta en marcha |  | La dificultad de la configuración es más compleja en Codeception, pero no supone una desventaja, ya que sus trabajadores son expertos en testing automático. |
| Complejidad para analizar los resultados |  | En este caso es más sencillo analizar los resultados. Dado que estos son más intuitivos, será más rápido analizar las conclusiones. |