## 2.1 Requisitos funcionales

Los requisitos funcionales deben ser los mismos para las dos implementaciones.

En la siguiente tabla se indicará el catálogo de requisitos funcionales del sistema:

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF01 | Efectuar pruebas para la operación suma |
| RF02 | Efectuar pruebas para la operación resta | |
| RF03 | Efectuar pruebas para la operación multiplicación | |
| RF04 | Efectuar pruebas para dividir | |
| RF05 | Controlar las posibles excepciones |

## 4.3 Documentación de pruebas

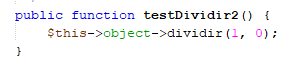
Se han establecido los siguientes casos de prueba para las funciones sumar, restar, multiplicar y dividir:

|  |  |
| --- | --- |
| **Primer número** | **Segundo número** |
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |
| 1 | 2 |

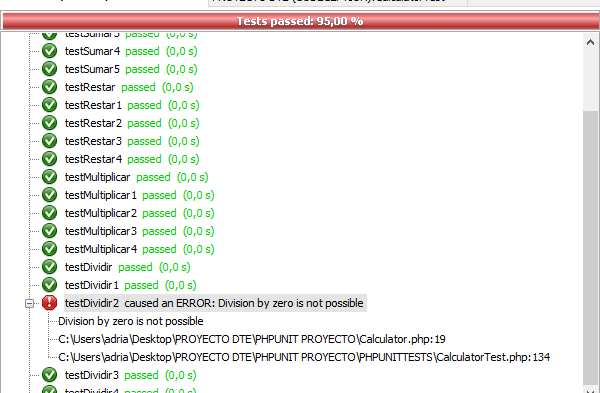
De esta manera, nos aseguramos de controlar resultados negativos, positivos y operaciones con el cero.

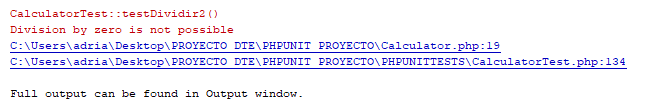
Para superar correctamente las pruebas, hay que indicar el resultado esperado de una operación. Esto ha sido correcto hasta que nos hemos encontrado con problemas en el momento que se da cuando se divide por cero:

En este ejemplo, se muestra el caso en el que se intenta dividir 1 entre 0:

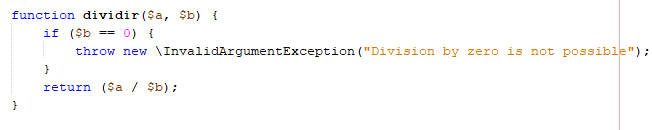


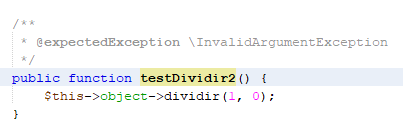
Por tanto, no se puede indicar el resultado esperado y se obtiene un error:





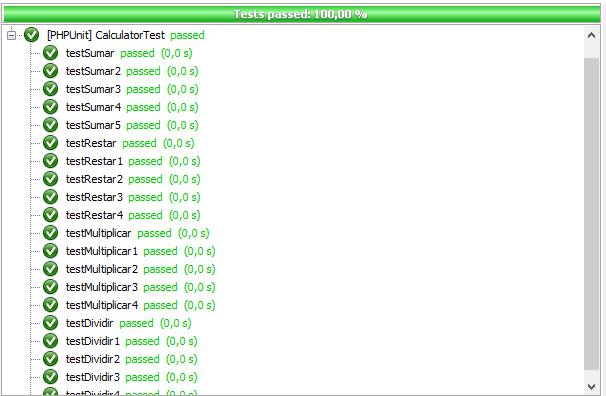
Para resolver esta situación, hay que controlar dicha excepción de la siguiente manera:





De esta forma, le estamos indicando que esperamos obtener una excepción al realizar esa operación. En concreto, la división por cero.

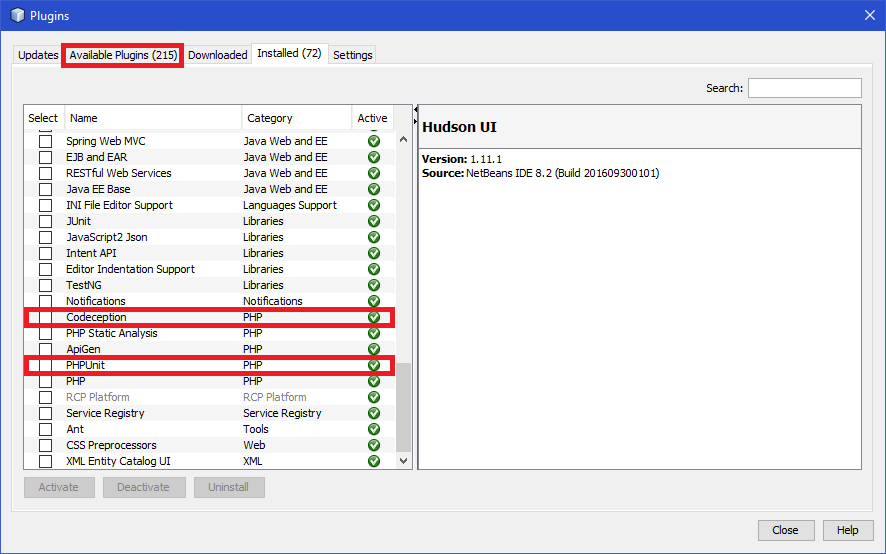
Finalmente, el resultado de las pruebas efectuadas es satisfactorio:

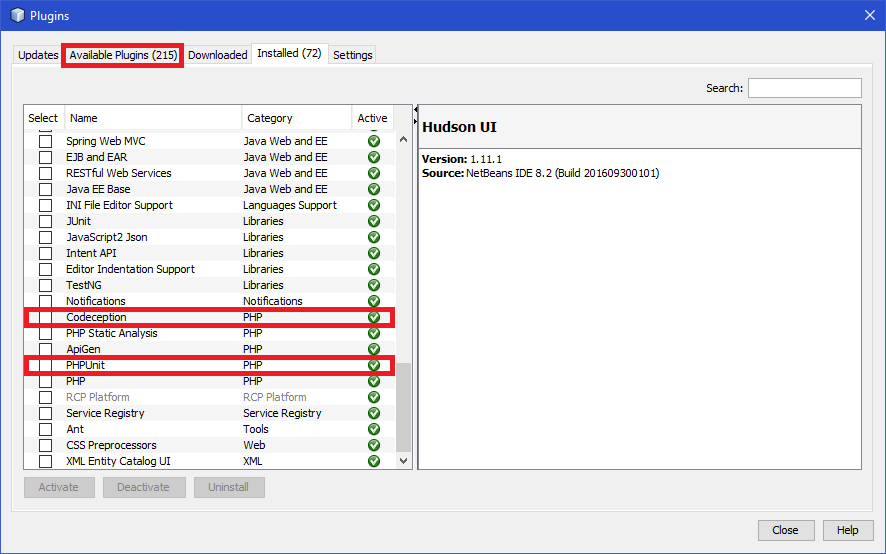


## 4.4 Documentación de instalación

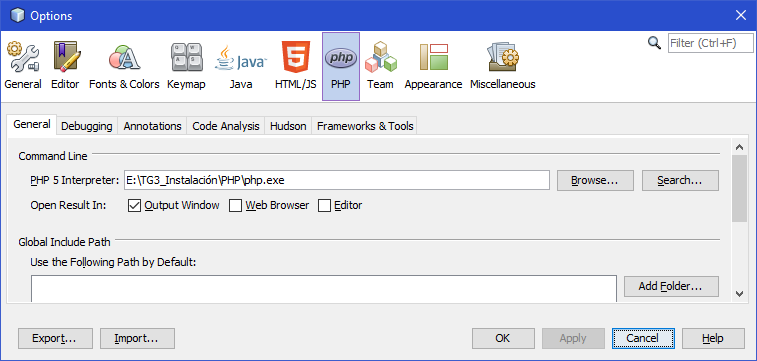
**Instalación de NetBeans**: en nuestro caso hemos utilizado NetBeans como plataforma para desarrollar los test con ambas plataformas. Se puede descargar la última versión desde el siguiente enlace, eligiendo la configuración completa (all) correspondiente a la última columna: <https://netbeans.org/downloads/8.2/>

**Plugin de PHP y PHPunit:** desde NetBeans hay que comprobar que estén instalados los plugins de PHP y PHPunit. Para ello hay que ir a la pestaña Tools y seleccionar Plugins. Es posible que si se ha elegido la configuración completa no sea necesario.

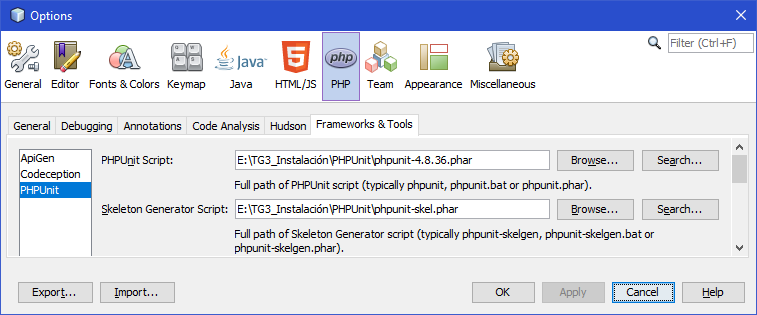




**Configurar PHP:** se accede a los ajustes de NetBeans en la pestaña Tools > Options. Hecho esto se accede al menú de PHP y en la pestaña General hay que seleccionar la ruta del archivo php.exe que se encuentra disponible en “TG3\_Instalación > PHP”, en el repositorio de GitHub (no está comprimido porque el tamaño supera el máximo permitido). Es necesario descargar la carpeta PHP completa ya que se han modificado los archivos .ini necesarios para que el proyecto funcione adecuadamente



**Configurar PHPunit con el archivo.phar y el Skeleton Generator:** se accede a los ajustes de NetBeans en la pestaña Tools > Options. Hecho esto se accede al menú de PHP y, a su vez, a la pestaña Frameworks & Tools, y se elige PHPUnit en el menú izquierdo.



En la primera caja (PHPUnit Script) hay que indicar la ruta al archivo PHAR de PHPUnit, y en el segundo el Skeleton Generator Script. Ambos se encuentran en la carpeta “TG3\_Instalación > PHPUnit” del repositorio de GitHub.

Por último, hay que descargar la carpeta del proyecto y abrirlo desde NetBeans.

**6. Comparación de las dos implementaciones**

Teniendo en cuenta los criterios de comparación definidos en el apartado 3 sobre la implementación de cada uno de los prototipos, se ha fijado la evaluación de los criterios en la implementación usando ambas herramientas.

**6.1 Evaluación de los criterios en la implementación usando PHPUnit**

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| C1 Dificultad instalación SW | 3 horas |
| C2 Recursos necesarios | 3 herramientas |
| C3 Tiempo de respuesta | 0,000 segundos |
| C4 Información disponible | 2 |
| C5 Implementación intuitiva | 2 |

**7. Comparación de la implementación de las tecnologías**

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERIOS | PHPUnit | Codeception | COMENTARIOS |
| C1 Dificultad instalación SW | 3 horas | 4 horas | Se ha tardado una hora más en Codeception, mayor complejidad. |
| C2 Recursos necesarios | 3 herramientas | 3 herramientas | Necesitamos la misma cantidad de herramientas- |
| C3 Tiempo de respuesta | 0,000 segundos | 0,007 segundos | En el código PHP creado se tarda más en realizarse las pruebas en Codeception, al ser un código tan pequeño los tiempos son mínimos, tanto es así, que en PHPUnit nos da 0,000 segundos. |
| C4 Información disponible | 2 | 1 | Se ha encontrado algo más de información en PHPUnit, pero aun así la cantidad de información encontrada ha sido muy reducida y difícil de encontrar. |
| C5 Implementación intuitiva | 2 | 1 | Como era más difícil encontrar la información en Codeception, ante un problema nos ha costado más resolverlo, y encima hemos tenido más problemas con Codeception para hacer funcionar sus tests. |

**8. Conclusiones**

Tal y como se explicó en el trabajo anterior PHPUnit y Codeception presentan funcionalidades similares.

Codeception es más aconsejable para aplicaciones grandes ya que la velocidad que ofrece para pruebas complejas es mayor. También ofrece una mayor usabilidad a la hora de entender los test de prueba. Sin embargo, el esfuerzo requerido para la puesta en marcha y la configuración es mucho mayor. Esto es debido a que tiene mayor complejidad, pero también a que hay mucha menos documentación disponible.

En este caso en concreto, la aplicación que se ha desarrollado es pequeña y los tests requeridos no implican una dificultad excesiva por lo que la mejor opción es PHPUnit.