Desarrollo Web en Entorno Servidor

5.- PHP - Librerías

Composer, Monolog, phpDocumentor, PHPUnit

IES Severo Ochoa

Ejercicio 501

ies severo ochoa

- Crea un nuevo proyecto con *Composer* llamado Monologos.
- Incluye como librería la ultima versión de *Monolog*.
- Crea la clase `Dwes\Monologos\HolaMonolog`
  - Define una propiedad privada para guardar el log
  - Define en el constructor un `RotatingFileHandler` que escribe en la carpeta `logs` del proyecto, y que almacene los mensajes a partir de debug.
  - Crea los métodos `saludar` y `despedir` que hagan un log de tipo info con la acción correspondiente.

Ejercicio 502

- Siguiendo con el proyecto Monologos:
  1) Crea un archivo llamado `inicio.php` que permita probar `HolaMonolog`.
  2) Comprueba que los mensajes aparecen en el log
  3) Cambia el nivel para que el manejador solo muestre los mensajes a partir de *warning*.
  4) Vuelve a ejectuar `inicio.php` y comprueba el archivo de log.

Ejercicio 503

- Modifica `HolaMonolog` y añade a la pila el manejador de *FirePHP* conjunto al procesador de Introspección, mostrando mensajes desde el nivel DEBUG.
- Añade una propiedad denominada `hora`, la cual se inicializa únicamente como parámetro del constructor
  - Si la hora es inferior a 0 o mayor de 24, debe escribir un log de warning con un mensaje apropiado.
- Modifica los métodos `saludar` y `despedir` para hacerlo acorde a la hora (buenos días, buenas tardes, hasta mañana, etc.)

Ejercicio 510

- Antes de nada, crea un repositorio privado en *GitHub* y sube el proyecto actual de *Videoclub*.

- Inicializa en local tu repostorio git → `git init`
- Añade y sube los cambios a tu repositorio → `git add .` y luego `git commit -m 'Inicializando proyecto'`.
- Renombra tu rama master a main → `git branch -M main`
- Conecta tu repositorio con GitHub y sube los cambios (mira la instrucciones de *GitHub*: comandos `git remote` y `git push`)

Ejercicio 510

ies severo ochoa

- Tras instalar *Composer*, inicialízalo dentro de tu proyecto *Videoclub*, y haz los cambios necesarios para utilizar el autoload de *Composer*.
  - Incluye Monolog y Phpunit, cada una en su lugar adecuado.
  - Añade el autoload al composer.json
- Sube los cambios a *GitHub* y crea la etiqueta v0.510

Ejercicio 511

- Vamos a modificar `Cliente` para sustituir poco a poco todos los usos del trait que teníamos por un `Logger` de *Monolog.*
- *Añade el log como una propiedad de la clase e inicializalo en el constructor*:
  - Nombre de canal: `VideoclubLogger`
  - Se debe almacenar en `logs/videoclub.log` mostrando todos los mensajes desde *Debug*.
  - Antes de lanzar cualquier excepción, debe escribir un log de tipo *Warning*.
  - Sustituya los usos que teníamos del trait (`logError` y `logWarning` por el respectivo método de *Monolog*).

Ejercicio 512

- Vuelve a hacer lo mismo que en el ejercicio anterior, pero ahora con la clase `Videoclub`.
- Además:
  - Los mensajes `logEcho` ahora serán info.
  - Eliminar el logger de `muestraResumen` para que vuelva a hacer `echo` de los mensajes. (eliminar la referencia al trait)
  - Siempre que se llame a un método de log, se le pasará como segundo parámetro la información que dispongamos.

- Ejecuta el archivo de prueba y comprueba que el log se rellena correctamente.

Ejercicio 513

- Vamos a refactorizar el código común de inicialización de *Monolog* que tenemos repetidos en los constructores a una factoría de Monolog:
  - `\Dwes\Videoclub\Util\LogFactory`
- Comprueba que sigue funcionando correctamente.
- Elimina todas las referencias al trait `Log`.

- Modifica la factoría para que devuelva `LogInterface` y comprueba su funcionamiento.
- Sube los cambios a *GitHub* y crea la etiqueta v0.513

Ejercicio 520

ies severo ochoa

- Instala phpdoc en tu sistema
- Ejecuta phpdoc sobre tu proyecto Monolog y comprueba el api que se crea.
- Comenta tanto la clase como los métodos.
- Vuelve a ejecutar phpdoc

Ejercicio 521

ies severo ochoa

- Ejecuta Doxygen sobre el proyecto modificado en el ejercicio anterio.
- Visualiza el resultado.

Ejercicio 530

- Documenta el proyecto Videoclub, y genera la documentación tanto con phpdoc como con Doxygen.
- Empieza por las clases de Soporte y sus hijos.
- Comprueba el resultado
- Luego sigue con Cliente y finalmente Videoclub.

Ejercicio 540

ies severo ochoa

- A partir de la clase `HolaMonolog`, modifica los métodos para que además de escribir en en log, devuelvan el saludo como una cadena.

- Crea la clase `HolaMonologTest` y añade diferentes casos de prueba para comprobar que los saludos y despedidas son acordes a la hora con la que se crea la clase.

Ejercicio 541

ies severo ochoa

- Vamos a simular TDD.
- Queremos que nuestra aplicación almacene los últimos tres saludos que ha realizado.
- Para ello:
  1) Crea las prueba necesarias (invoca al método saludar/despedir varias veces y llama al método que te devuelva los saludos almacenados)
  2) Implementa el código para pasar las pruebas
  3) Refactoriza el código

Ejercicio 542

- Utiliza proveedores de datos para comprobar esta última funcionalidad, pasándole:
  - Ningún saludo / despedida.
  - Un saludo / despedida.
  - Tres saludos / despedidas.
  - Cuatro saludos / despedidas.
  - Combinaciones de saludos / despedidas.

Ejercicio 543

- ¿Recuerdas que si la hora es negativa o superior a 24 escribíamos en el log un warning?
- Ahora debe lanzar una `InvalidArgumentException`
- Vuelve a aplicar TDD y completa tus casos de prueba.

Ejercicio 544

- Comenta la última prueba realizada (la comprobación de las excepciones) y realiza un informe de cobertura de pruebas.
- Analiza los resultados obtenidos.
- Elimina los últimos comentarios sobre la última prueba y vuelve a generar y analizar el informe de cobertura.

Ejercicio 550

- El objetivo de los siguientes ejercicios es conseguir de manera incremental una cobertura de pruebas superior al 95%.
- Crea pruebas dentro de la carpeta `tests` para las clases `Soporte`, `CintaVideo`, `Dvd` y `Juego`.
  - Recuerda respetar el espacio de nombres.
  - Los métodos `muestraResumen`, tras hacer `echo` de los mensajes, deben devolver una cadena con el propio mensaje.

Ejercicio 551

- Crea pruebas para la clase `Cliente`
- Aprovecha todo el código que teníamos para comprobar la funcionalidad.
- Utiliza proveedores de datos para añadir conjuntos de datos mayores que los empleados
  - Comprueba que funciona con diferentes cupos, que al intentar alquilar un soporte marcado como ya alquilado debe lanzar una excepción, que no coincidan los ids de los soportes, etc...

Ejercicio 552

- Crea las pruebas para la clase `Videoclub`

- Ten en cuenta los últimos métodos añadidos que permitían alquilar y devolver soportes, tanto de manera individual como mediante un array.

Ejercicio 553

ies severo ochoa

- Crea el informe de cobertura.
- Analiza los datos de cobertura (>= 90%) y comprueba el valor de CRAP, de manera que siempre sea <= 5.
- En caso de no cumplirse, crea nuevos casos de prueba y/o refactoriza el código de tu aplicación.

- Sube los cambios a *GitHub* y crea la etiqueta v0.533

Ejercicio 554

- Queremos que en `Videoclub`, cuando un cliente no existe (tanto al alquilar como al devolver) se lance una nueva excepción: `ClienteNoExisteException`.
- Además, dado el número creciente de excepciones, queremos mover las excepciones al namespace `Dwes\Videoclub\Exception`
- Siguiendo TDD, primero crea las pruebas, y luego modifica el código de aplicación.
- Vuelve a generar el informe de cobertura y comprueba la calidad de nuestras pruebas.
- Sube los cambios a *GitHub* y crea la etiqueta v0.544

Ejercicio 555

- ¿Nadie se ha dado cuenta que en los `Dvd` no estamos almacenando su duración?
- Haz todos los cambios necesarios, primero en las pruebas y luego en el código.

- Sube los cambios a *GitHub* y crea la etiqueta v0.545

Ejercicio 556

ies severo ochoa

- Tras años luchando contra la tecnología, decidimos introducir los Blu-ray en nuestra empresa.
  - Hemos decido que `Bluray` herede de `Soporte`.
  - Además del título y la duración, nos interesa almacenar si es 4k.
- Haz todos los cambios necesarios, primero en las pruebas y luego en el código.

- Sube los cambios a *GitHub* y crea la etiqueta v0.546

¿Alguna pregunta?