

## ANÁLISIS DE LOS CASOS DE PRUEBA:

-domain\_with\_trailing\_slash\_when\_construction\_an\_api\_path

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámica**: Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra**: Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para conocer el resultado que devuelve.
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional**: Se comprueba el funcionamiento de la aplicación

-domain\_without\_trailing\_slash\_when\_construction\_an\_api\_path

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámica**: Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra**: Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para considerar la posibilidad de introducir un dominio sin barra al final.
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional**: Se comprueba el funcionamiento de la aplicación.

-query\_string\_from\_map

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámica**: Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra**: Se testea que el resultado de un método publico de la clase es correcto. Es necesario unicamente conocer la interfaz del método.
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional**: Se comprueba el funcionamiento de la aplicación.

-url\_encoded\_query\_string\_from\_map

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámicas**: Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra**: Se testea que el resultado de un método publico de la clase es correcto. No es necesario conocer la implementación interna del método para considerar la posibilidad de formar un string con caracteres especiales probando así el encoding.
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional**: Se comprueba el funcionamiento de la aplicación.

-api\_url\_need\_to\_be\_absolute

- **Test de unidad :** Testan unicamente la clase de EPLiteConnection
- **Dinámicas:** Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra:** Se testea que el resultado de un método publico de la clase es correcto. Para tener en cuenta la posibilidad de probar el método con rutas relativas y absolutas no implica conocer la implementación del método.
- **Negativa:** Se comprueba el correcto funcionamiento del método cuando las entradas generan una excepción.
- **Funcional:** Se comprueba el funcionamiento de la aplicación.

-handle\_valid\_response\_from\_server

- **Test de unidad :** Testan unicamente la clase de EPLiteConnection
- **Dinámicas:** Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra:** Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para conocer el resultado que devuelve.
- **Positiva :** Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional:** Se comprueba el funcionamiento de la aplicación

-handle\_invalid\_parameter\_error\_from\_server

- **Test de unidad :** Testan unicamente la clase de EPLiteConnection
- **Dinámicas:** Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra:** Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para conocer el resultado que devuelve.
- **Negativa:** Se comprueba el correcto funcionamiento del método cuando las entradas generan una excepción.
- **Funcional:** Se comprueba el funcionamiento de la aplicación

-handle\_internal\_error\_from\_server

- **Test de unidad :** Testan unicamente la clase de EPLiteConnection
- **Dinámicas:** Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra:** Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para conocer el resultado que devuelve.
- **Negativa:** Se comprueba el correcto funcionamiento del método cuando las entradas generan una excepción.
- **Funcional:** Se comprueba el funcionamiento de la aplicación

#### -handle\_no\_such\_function\_error\_from\_server

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para conocer el resultado que devuelve.
- **Negativa** : Se comprueba el correcto funcionamiento del método cuando las entradas generan una excepción.
- **Funcional** : Se comprueba el funcionamiento de la aplicación

#### -handle\_invalid\_key\_error\_from\_server

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para conocer el resultado que devuelve.
- **Negativa** : Se comprueba el correcto funcionamiento del método cuando las entradas generan una excepción.
- **Funcional** : Se comprueba el funcionamiento de la aplicación

#### -unparsable\_response\_from\_the\_server

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para conocer el resultado que devuelve.
- **Negativa** : Se comprueba el correcto funcionamiento del método cuando las entradas generan una excepción.
- **Funcional** : Se comprueba el funcionamiento de la aplicación

#### -unexpected\_response\_from\_the\_server

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Se testea que el resultado de un método publico de la clase es correcto, no es necesario conocer la estructura interna de este para conocer el resultado que devuelve.
- **Negativa** : Se comprueba el correcto funcionamiento del método cuando las entradas generan una excepción.
- **Funcional** : Se comprueba el funcionamiento de la aplicación
- valid\_response\_with\_null\_data

-valid\_response\_with\_null\_data

- **Test de unidad** : Testan unicamente la clase de EPLiteConnection
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Se testea que el resultado de un método publico de la clase es correcto. Para tener en cuenta la posibilidad de que los datos de la respuesta sean nulos no es necesario conocer la implementación.
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

-validate\_token ( El cliente solicita el token y comprueba que sea correcto)

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Unicamente se hace una llamada al método para lo cual es unicamente necesario conocer su firma. No se realiza ninguna otra acción que implique conocer la implementación concreta del método.
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.
- 

-create\_and\_delete\_group ( Crea un grupo y comprueba que la respuesta es la correcta)

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) la respuesta es recibida tal y como la envía el servidor por lo que la implementación concreta de los métodos no hace variar el formato de esta.
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

-create\_group\_if\_not\_exists\_for\_and\_list\_all\_groups

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) .
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

-create\_group\_pads\_and\_list\_them

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) .
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

-create\_author

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) .
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

-create\_author\_with\_author\_mapper

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) .
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

-create\_and\_delete\_session/

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) . En el caso del método createSession se le puede pasar como parametro la duración de la session o la fecha de fin de esta, el tipo de valores lo podemos encontrar en la firma del método y la unidad de medida que simbolizan aparece representada en el javadoc por lo que no necesitamos consultar la implementación para este test.
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

-create\_pad\_set\_and\_get\_content

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) .
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

-create\_pad\_move\_and\_copy

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) .
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas
- **Funcional** : Se comprueba el funcionamiento de la aplicación

-create\_pads\_and\_list\_them

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) .
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación

-create\_pad\_and\_chat\_about\_it

- **Test de Integración** : Testea la comunicación entre varios componentes
- **Dinámicas** : Requiere de la ejecución de la aplicación para ejecutarse
- **Caja negra** : Para la llamada a los métodos de la clase EPLiteClient unicamente es necesario conocer su firma ( no su implementación) .
- **Positiva** : Se comprueba el correcto funcionamiento del método cuando las entradas son correctas.
- **Funcional** : Se comprueba el funcionamiento de la aplicación.

## **CONCLUSIONES:**

Como se puede ver en el análisis de los test, los casos de prueba realizados son de caja negra lo que supone que no se tienen en cuenta las implementaciones concretas de los métodos , de forma que no se esta probando la calidad del software desarrollado sino mas bien su carácter funcional. Directamente relacionado podemos ver que los test son en su totalidad funcionales y por lo tanto no se comprueban aspectos relacionados con el rendimiento del software ( pruebas no funcionales).

E cuanto a la pruebas unitarias se prueban todos los métodos y atributos, sin embargo, no se hace una prueba exhaustiva sobre estos, puesto que no se hacen test con valores aleatorios ni con valores frontera, lo que aumentaría la confianza sobre las pruebas.

Por su parte las pruebas de integración, testean todas las interacciones entre los componentes, sin embargo, al ser en su mayoría positivas, no se verifica el correcto funcionamiento del sistema ante entradas erróneas o comportamientos inesperados.

En definitiva, aunque el código se encuentra en su gran mayoría testeado, es decir, la mayor parte de las líneas de código son ejecutadas por los test, estos solo comprueban en su mayoría que ese código se ejecute correctamente en situaciones regulares y con valores normales ( acercándose así a las pruebas de aceptación ) , lo que implica que se podrían realizar una gran variedad de tests adicionales que podrían resultar satisfactorios ( que encuentren errores ).

Christian Romarís Caamaño