

Para empezar con un primer filtrado de los test los podemos dividir en Test de Unidad y Test de integración, con 13 test en el haber del primer grupo y 11 en el segundo.

PRUEBAS UNITARIAS

Comenzando con la clasificación y definición de las pruebas de unidad nos encontramos con una clase llamada “EPLiteConnectionTest” que alberga test de unidad bastante completos sobre la clase “EPLiteConnection”. Por otra parte, existen dos clases como son “GETRequest” y “POSTRequest” que carecen de cualquier tipo de prueba unitaria, y aunque su implementación resulte trivial, es totalmente necesario. Además existe la clase “EPLiteClient” que aunque se prueba en los test de integración, no posee test a nivel unitario independientes de la clase “EPLiteConnection”.

◆ Listado de pruebas facilitadas en la clase “EPLiteConnectionTest” con su respectivo análisis:

- `domain_with_trailing_slash_when_construction_an_api_path()` :
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso se intenta forzar un comportamiento dentro del código intentando pasarle a la url un “slash” al final para que la aplicación reaccione de manera diferente, por lo que se trataría de una prueba de caja blanca, ya que la persona que realiza la prueba tiene cierto conocimiento del código para el diseño de la misma. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parámetros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `domain_without_trailing_slash_when_construction_an_api_path()`:
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso se intenta forzar un comportamiento dentro del código intentando pasarle al método requerido la misma url que en el caso de prueba anterior pero esta vez si el “slash” al final para que la aplicación reaccione de manera diferente, por lo que se trataría de una prueba de caja blanca, ya que la persona que realiza la prueba tiene cierto conocimiento del código para el diseño de la misma. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.

- `query_string_from_map()`

- En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.

- `url_encoded_query_string_from_map()`

- En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.

- `api_url_need_to_be_absolute()`

- En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el primer grupo pues en su naturaleza está la comprobación del correcto funcionamiento de los métodos ante un mal uso de los mismos, intentando de esta manera que se lancen excepciones o se muestren mensajes de error.

- `handle_valid_response_from_server()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `handle_invalid_parameter_error_from_server()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el primer grupo pues en su naturaleza está la comprobación del correcto funcionamiento de los métodos ante un mal uso de los mismos, intentando de esta manera que se lancen excepciones o se muestren mensajes de error.
- `handle_internal_error_from_server()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el primer grupo pues en su naturaleza está la comprobación del correcto funcionamiento de los métodos ante un mal uso de los mismos, intentando de esta manera que se lancen excepciones o se muestren mensajes de error.

- `handle_no_such_function_error_from_server()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el primer grupo pues en su naturaleza está la comprobación del correcto funcionamiento de los métodos ante un mal uso de los mismos, intentando de esta manera que se lancen excepciones o se muestren mensajes de error.
- `handle_invalid_key_error_from_server()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el primer grupo pues en su naturaleza está la comprobación del correcto funcionamiento de los métodos ante un mal uso de los mismos, intentando de esta manera que se lancen excepciones o se muestren mensajes de error.
- `unparsable_response_from_the_server()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el primer grupo pues en su naturaleza está la comprobación del correcto funcionamiento de los métodos ante un mal uso de los mismos, intentando de esta manera que se lancen excepciones o se muestren mensajes de error.

- `unexpected_response_from_the_server()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el primer grupo pues en su naturaleza está la comprobación del correcto funcionamiento de los métodos ante un mal uso de los mismos, intentando de esta manera que se lancen excepciones o se muestren mensajes de error.
- `valid_response_with_null_data():`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que en este caso esta concretamente se clasificaría como una prueba de tipo caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.

******La clasificación en pruebas de tipo caja negra y caja blanca puede ser difuso en ciertas ocasiones por el simple hecho de que la persona que realizó las pruebas es la misma que implementó el código de la aplicación. Esto implica que su conocimiento del código puede influir en la realización de las mismas.

******En las pruebas de tipo negativo en todos los casos se captura (o se espera con la anotación `@Test(expected =...)`) la excepción propia de la aplicación “EPLiteException”.

PRUEBAS DE INTEGRACIÓN:

Continuando con la clasificación antes mencionada nos encontramos con los test de Integración. Para este tipo de test se ha definido en la aplicación la clase “EPLiteClientIntegrationTest” donde se realizan test donde se prueba la integración entre la clase “EPLiteConnection” y “EPLiteClient”.

****NO SE HAN ENCONTRADO USOS DE LAS CLASES GETRequest y POSTRequest así como tampoco se han encontrado test que integren su uso con el resto de componentes.**

◆ Las pruebas en cuestión a analizar será el siguiente conjunto de 11 tests:

- `validate_token()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `create_and_delete_group()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.

- `create_group_if_not_exists_for_and_list_all_groups()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `create_group_pads_and_list_them()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `create_author()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.

- `create_author_with_author_mapper()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `create_and_delete_session()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `create_pad_set_and_get_content()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.

- `create_pad_move_and_copy()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `create_pads_and_list_them()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.
- `create_pad_and_chat_about_it()`
 - En primer lugar, en referencia a la clasificación de pruebas en caja negra y caja blanca, podemos observar que esta podría clasificarse como de caja negra, esto se debería a que su implementación no depende del conocimiento del código de antemano, solo es necesario saber el resultado que se pretende obtener. Con respecto a su clasificación en pruebas de tipo dinámico, estático o simbólico esta se englobaría dentro del tipo dinámico ya que para comprobar la respuesta que la prueba necesita para verificar su funcionamiento será necesario en todos los casos la ejecución del software. A continuación, pasando a las pruebas de tipo funcional, estructural o no funcional podemos identificar esta como de tipo funcional, deduciendo esto ya que en su esencia buscan probar el funcionamiento del código en si, sin comprobar su estructura ni su capacidad de carga (entre otros parametros de rendimiento). Finalmente, intentando caracterizarla en función de si es una prueba negativa o positiva podemos clasificarla en el segundo grupo ya que su objetivo es probar el correcto funcionamiento de cierto método intentando que este responda de manera correcta, nunca buscando el lanzamiento de excepciones o mensajes de error.

CONCLUSIÓN:

Como ya se ha indicado anteriormente existe una carencia de pruebas a nivel unitario ya que existen clases que no poseen pruebas de este tipo (GETRequest", "POSTRequest" y "EPLiteClient").

Por otra parte, en lo referente a las pruebas de integración no encontramos ningún uso de las clases "GETRequest" y "POSTRequest" en los test de integración proporcionados. Además, los test de integración que están a nuestra disposición buscan siempre el funcionamiento correcto de los métodos, es decir carece de pruebas de tipo negativo.

Con respecto al nivel de cobertura obtenido, se puede observar que existe aproximadamente un 6% de código que no está siendo probado, en este caso en la clase "EPLiteConnection" donde en diferentes partes del código no se pasa por las ramas que encapsulan el comportamiento anómalo de los métodos mediante la captura de excepciones. Además, en la clase "EPLiteClient" no se prueba la construcción del objeto con uno de sus constructores.

Finalmente y de manera general, cabe destacar que la aplicación no posee ningún test de tipo estructural, no funcional, estático ni simbólico.