

Pruebas unitarias productos:

Pruebas del modelo

Para probar el modelo y así comprobar que la comunicación con la base de datos está bien se hicieron dos test en el modelo.

Estos fueron, el primero, sirve para probar que al pedir todos los productos de la base de datos efectivamente se retornen todos los registros que deberían estar en la base de datos.

Para lograr probarlo se creó la función getAllProducts() en el modelo, dicha función es la siguiente:

```
public function getAllProducts() {
    return $this->find('all', array(
        'fields' => array('id', 'name',
            'category', 'type', 'price',
            'weight', 'filename', 'dir',
            'created', 'keywords', 'volumen')
        ));
}
```

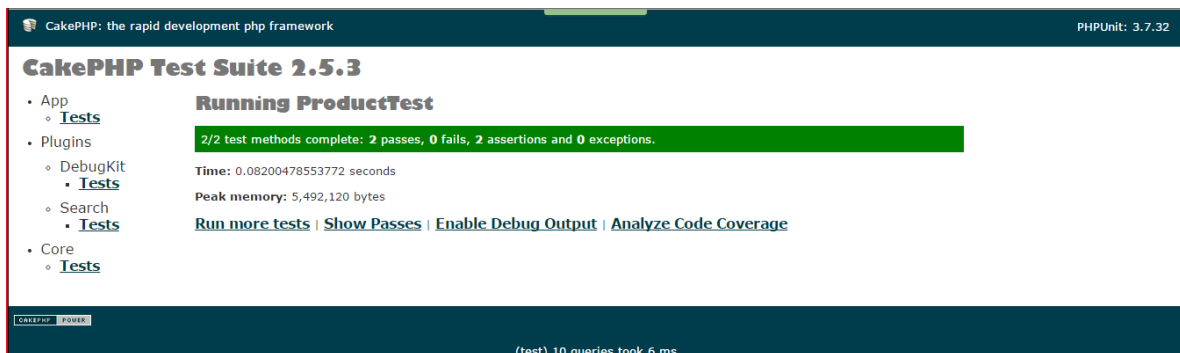
1Función GetAllProducts

El segundo sirve para probar si efectivamente las condiciones de consulta se hacen correctamente desde el modelo, para esto se creó la función getProductsFiltrados(\$id), el cual por medio del id, llave primaria, como criterio de consulta, se restringe la devolución de registros. La función getProductsFiltrados(\$id) es la siguiente:

```
public function getProductsFiltrados($id = null) {
    return $this->find('first', array(
        'conditions' => array('id' => $id)
    ));
}
```

2Función GetProductsFiltrado

El resultado de ejecutar las pruebas del modelo se pueden observar en la figura 3.



The screenshot displays the CakePHP Test Suite 2.5.3 interface. At the top, it says 'CakePHP: the rapid development php framework' and 'PHPUnit: 3.7.32'. The main heading is 'CakePHP Test Suite 2.5.3'. On the left, there is a sidebar with a tree view showing 'App' (with 'Tests' selected), 'Plugins', 'DebugKit' (with 'Tests' selected), 'Search' (with 'Tests' selected), and 'Core' (with 'Tests' selected). The main content area is titled 'Running ProductTest'. It shows a green progress bar indicating '2/2 test methods complete: 2 passes, 0 fails, 2 assertions and 0 exceptions.' Below this, it lists 'Time: 0.08200478553772 seconds' and 'Peak memory: 5,492,120 bytes'. At the bottom of the main content area, there are links: 'Run more tests', 'Show Passes', 'Enable Debug Output', and 'Analyze Code Coverage'. The footer of the interface shows '(test) 10 queries took 6 ms'.

3Resultado del Test Modelo Producto

Ahora si se observa la impresión de las consultas de la base de datos de la figura 4 se notan los dos select que se hacen el primero que pide todos los registros y el segundo que está restringido con respecto al id = 3. Que corresponden a los llamados de las funciones getAllProducts y getProductsFiltardos

5	SELECT 'Product'.id, 'Product'.name, 'Product'.category, 'Product'.type, 'Product'.price, 'Product'.weight, 'Product'.filename, 'Product'.dir, 'Product'.created, 'Product'.keywords, 'Product'.volumen FROM 'test'.products AS 'Product' WHERE 1 = 1	2	2	1
6	BEGIN	2	2	1
7	INSERT INTO 'test'.products ('id', 'name', 'category', 'type', 'price', 'weight', 'filename', 'dir', 'created', 'keywords', 'volumen') VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?), params[3, Balon Barcelona, Futbol, Balon del FCBarcelona, 13.5, 15, bolaBarca.jpg, img/uploads/product/filename, 2014-10-16 15:56:23, futbol, NULL]	2	2	1
8	INSERT INTO 'test'.products ('id', 'name', 'category', 'type', 'price', 'weight', 'filename', 'dir', 'created', 'keywords', 'volumen') VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?), params[4, Balon Nike, Futbol, Balon del FCNIke, 13.5, 15, bolaNIke.jpg, img/uploads/product/filename, 2014-10-16 15:56:23, nike, NULL]	2	2	1
9	COMMIT	2	2	1
10	SELECT 'Product'.id, 'Product'.name, 'Product'.category, 'Product'.type, 'Product'.price, 'Product'.weight, 'Product'.filename, 'Product'.dir, 'Product'.created, 'Product'.keywords, 'Product'.volumen FROM 'test'.products AS 'Product' WHERE 'id' = 3 LIMIT 1	1	1	1

Pruebas del Controlador

Para probar el controlador de la clase Products se hicieron 7 tests, de los cuales todos se aprobaron. Dichos tests son los siguientes:

1. testIndex: Se utiliza para saber si el index está devolviendo todos los registros a la vista correspondiente. En la siguiente imagen se puede observar como se mantiene un resultado esperado, luego se llama a la función index y este devolverá todos los registros de la base de datos entonces por esta razón se comparan entre lo esperado con el primer registro insertado en la base de datos, ya que este y en el esperado son el mismo, por lo tanto se comparan para probar que si lo devuelve como tiene que ser.

```
public function testIndex() {

    $result1 = $this->testAction('/products/index');

    $expected = array(
        'Product' => array(
            'id' => 3,
            'name' => 'Balon Barcelona',
            'category' => 'Futbol',
            'type' => 'Balon del FCBarcelona',
            'price' => 13.500,
            'weight' => '15',
            'filename' => 'bolaBarca.jpg',
            'dir' => 'img/uploads/product/filename',
            'created' => '2014-10-16 15:56:23',
            'keywords' => 'futbol',
            'volumen' => 'NULL')
    );

    $result = Hash::extract($this->vars['products'], '{n}');
    $this->assertEquals($expected, $result[0]);

}
```

2. testSearchDelete: Se usa para probar que efectivamente devuelve el registro que se desea borrar.

```
public function testSearchDelete() {

    $result = $this->testAction('/products/searchDelete/3');

    $expected = array(
        'Product' => array(
            'id' => 3,
            'name' => 'Balon Barcelona',
            'category' => 'Futbol',
            'type' => 'Balon del FCBarcelona',
            'price' => 13.500,
            'weight' => '15',
            'filename' => 'bolaBarca.jpg',
            'dir' => 'img\uploads\product\filename',
            'created' => '2014-10-16 15:56:23',
            'keywords' => 'futbol',
            'volumen' => 'NULL')
    );

    $this->assertEquals($this->vars['products'], $expected);

} // fin de testSearchDelete()
```

3. testDelete: Elimina el registro que SearchDelete devuelve por lo tanto en la impresión de consultas se puede observar que se borra el registro con id=3.

```
public function testDelete() {

    $expected = array(
        'Product' => array(
            'id' => 3,
            'name' => 'Balon Barcelona',
            'category' => 'Futbol',
            'type' => 'Balon del FCBarcelona',
            'price' => 13.500,
            'weight' => '15',
            'filename' => 'bolaBarca.jpg',
            'dir' => 'img\uploads\product\filename',
            'created' => '2014-10-16 15:56:23',
            'keywords' => 'futbol',
            'volumen' => 'NULL')
    );

    $this->testAction('/products/delete/', array('data' => $expected, 'method' => 'post'));

} // fin de testDelete
```

```
19 DELETE `Product` FROM `test`.`products` AS `Product` WHERE `Product`.`id` = 3
```

4. testSearchUpdate: Se usa para probar que efectivamente devuelve el registro que se desea actualizar.

```
public function testSearchUpdate() {  
  
    $result = $this->testAction('/products/searchUpdate/3');  
  
    $expected = array(  
        'Product' => array(  
            'id' => 3,  
            'name' => 'Balon Barcelona',  
            'category' => 'Futbol',  
            'type' => 'Balon del FCBarcelona',  
            'price' => 13.500,  
            'weight' => '15',  
            'filename' => 'bolaBarca.jpg',  
            'dir' => 'img\uploads\product\filename',  
            'created' => '2014-10-16 15:56:23',  
            'keywords' => 'futbol',  
            'volumen' => 'NULL')  
        );  
  
    $this->assertEquals($this->vars['products'], $expected);  
  
} // fin de testSearchUpdate()
```

5. testUpdate: Elimina el registro que SearchUpdate devuelve por lo tanto en la impresión de consultas se puede observar que se actualiza el registro con id=3.

```
34 UPDATE `test`.`products` SET `id` = 3, `name` = 'Balon', `category` = 'Futbol', `type` = 'Champions League', `price` = 12000, `weight` = '2', `dir` =  
    'img\uploads\product\filename', `created` = '2014-10-16 15:56:23', `keywords` = 'Futbol', `volumen` = 'NULL' WHERE `test`.`products`.`id` = 3
```

6. testadd: Se utiliza para saber que efectivamente se inserta un producto en la base de datos, haciendo uso de la función add.

```
public function testAdd() {  
    $insercion =  
        array(  
            'Product' => array(  
                'name' => 'Barca',  
                'category' => 'Futbol',  
                'dir' => 'NULL',  
                'type' => 'Balon Nike',  
                'price' => 2326,  
                'weight' => '2323',  
                'volumen' => '2323',  
                'keywords' => 'bola')  
            );  
}
```

7. testBusqueda: Se creo una función búsqueda() la cual sirve para demostrar que al mandar un criterio de búsqueda al index este me envía solo los registros que cumplen este criterio.

```
public function testbusqueda() {  
  
    $this->testAction('/products/index?keywords=futbol&type=Balon del FCBarcelona');  
  
    $expected = array(  
        'Product' => array(  
            'id' => 3,  
            'name' => 'Balon Barcelona',  
            'category' => 'Futbol',  
            'type' => 'Balon del FCBarcelona',  
            'price' => 13.500,  
            'weight' => '15',  
            'filename' => 'bolaBarca.jpg',  
            'dir' => 'img/uploads/product/filename',  
            'created' => '2014-10-16 15:56:23',  
            'keywords' => 'futbol',  
            'volumen' => 'NULL')  
        );  
    pr($this->vars['products']);  
    $result = Hash::extract($this->vars['products'], '{n}');  
  
    $this->assertEquals($expected, $result[0]);  
}
```

Si se analiza las impresiones de las consultas se puede observar como efectivamente se restringe el select, que devuelve los registros al index. En la figura de abajo se puede ver como se seleccionan los registros que cumplen con el like.

```
46 SELECT `Product`.`id`, `Product`.`name`, `Product`.`category`, `Product`.`type`, `Product`.`price`, `Product`.`weight`, `Product`.`filename`,  
  `Product`.`dir`, `Product`.`created`, `Product`.`keywords`, `Product`.`volumen` FROM `test`.`products` AS `Product` WHERE `Product`.`keywords` LIKE  
  '%futbol%' AND `Product`.`type` LIKE '%Balon del FCBarcelona%' LIMIT 20
```

El resultado de efectuar estas pruebas es el siguiente:

CakePHP Test Suite 2.5.3

- App
 - [Tests](#)
- Plugins
 - DebugKit
 - [Tests](#)
 - Search
 - [Tests](#)
- Core
 - [Tests](#)

Running ProductsControllerTest

7/7 test methods complete: 7 passes, 0 fails, 5 assertions and 0 exceptions.

Time: 0.50402903556824 seconds

Peak memory: 9,587,592 bytes

[Run more tests](#) | [Show Passes](#) | [Enable Debug Output](#) | [Analyze Code Coverage](#)