

Curso de Desarrollo Web en PHP orientado a objetos con MVC

Eugenia Bahit, Abril 2015

12

Sustituciones iterativas
expresiones regulares y otros temas

1

Crear plantilla HTML

```
<table id="listado" class="hover stripe table-responsive">
  <thead>
    <tr>
      <th>ID</th>
      <th>Producto</th>
      <th class="text-center">Precio</th>
      <th>&nbsp;</th>
    </tr>
  </thead>
  <tbody>

    <!--listado-->
    <tr>
      <td>{producto_id}</td>
      <td><a href="/producto/ver?id={producto_id}">{denominacion}</a></td>
      <td class="text-center">${ {precio} }.</td>
      <td class="text-center">
        <a href="/producto/editar?id={producto_id}"><i class="fa fa-edit"></i></a> |
        <a href="/producto/eliminar?id={producto_id}"><i class="fa fa-trash-o"></i></a>
      </td>
    </tr>
    <!--listado-->

  </tbody>
</table>

<script type="text/javascript">
  $(document).ready(function() {
    $('#listado').dataTable({
      language: {
        url: '//cdn.datatables.net/plug-ins/3cfcc339e89/i18n/Spanish.json'
      }
    });
  });
</script>
```

2

Crear función listar() en la vista

```
function listar($coleccion) {
    # Obtener plantillas
    $plantilla = file_get_contents('static/listar_productos.html');

    # Obtener HTML sobre el cual iterar
    $regex = "/<!--listado-->(.\|\n){1,}<!--listado-->/";
    preg_match($regex, $plantilla, $coincidencias);
    $html = $coincidencias[0];

    # Realizar sustitución iterativa
    $iter = '';
    foreach($coleccion as $obj) {
        $claves = array_keys(get_object_vars($obj));
        foreach($claves as &$clave) $clave = '{' . $clave . '}';
        $values = array_values(get_object_vars($obj));
        $iter .= str_replace($claves, $values, $html);
    }

    # Finalizar sustitución
    $contenido = str_replace($html, $iter, $plantilla);

    # Retornar título y contenido
    Return array('Listado de productos', $contenido);
}
```

3

Modificar recurso listar() del controlador

```
function listar() {  
    $colector = new Collector();  
    $colector->get('Producto');  
    return $this->view->listar($colector->coleccion);  
}
```

¡ATENCIÓN!

Si se experimenta un funcionamiento anormal del recurso, agregar las siguientes líneas al comienzo y al final de la **función listar() de la vista**, respectivamente:

```
# Al comienzo de la función  
$pcre_limit = ini_set("pcre.recursion_limit", 10000);  
  
# Al final de la función ANTES del return  
ini_set("pcre.recursion_limit", $pcre_limit);
```