

# Manual de MVC: (2)

## Vistas dinámicas y

# Templates

Tanto en Python como en PHP, es posible no embeber diferentes lenguajes en las plantillas. Incluso, cuando se trate de resultados traídos de una base de datos, que deban ser sustituidos de forma iterativa. En la entrega de hoy, aprenderemos las técnicas que nos convertirán en “Magos del MVC”.

Escrito por: **Eugenia Bahit** (Arquitecta GLAMP & Agile Coach)



Eugenia es **Arquitecta de Software**, **docente** instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y eXtreme Programming. Miembro de la [Free Software Foundation](#) e integrante del equipo de [Debian Hackers](#).

**Webs:**

Cursos de programación a Distancia: [www.cursosdeprogramacionadistancia.com](http://www.cursosdeprogramacionadistancia.com)  
Agile Coaching: [www.eugeniabahit.com](http://www.eugeniabahit.com)

**Redes sociales:**

Twitter / Identi.ca: [@eugeniabahit](#)

Creo que la consulta más frecuente que he recibido desde que publiqué mi libro «*POO y MVC en PHP*», ha sido “cómo evitar embeber código de programación en el HTML para completar una tabla o lista de selección (`select`) con los resultados de una consulta a base de datos”. Y créanme: es sumamente sencillo.

Primero, es necesario saber, que existen tres tipos de sustituciones que pueden efectuarse en las vistas:

1. **Sustituciones estáticas:** es el caso de una plantilla HTML en la que se deben sustituir ciertos datos de forma estática. Es decir, plantillas HTML en las que cada indicador de sustitución, debe ser reemplazado por un valor concreto.
2. **Sustituciones dinámicas:** son aquellas plantillas HTML en las cuáles un mismo indicador de sustitución, debe ser reemplazado de forma iterativa por más de un dato. Por ejemplo, una lista de selección que deba reemplazarse con los resultados devueltos por una consulta de selección múltiple a base de datos.

3. **Sustituciones combinadas:** son aquellas plantillas HTML en las que se debe realizar tanto una sustitución estática como una (o más) sustituciones dinámicas.

Pasemos lo anterior a ejemplos prácticos:

#### Plantilla de sustitución estática:

```
<!doctype html>
<html lang="es">
  <head>
    <charset="utf-8">
    <title>[INDICADOR_DE_SUSTITUCION: TITULO]</title>
  </head>

  <body>
    <header>
      <h1>[INDICADOR_DE_SUSTITUCION: TITULO]</h1>
    </header>

    <article>
      [INDICADOR_DE_SUSTITUCION: CONTENIDO]
    </article>
  </body>
</html>
```

#### Plantilla de sustitución dinámica:

```
<!doctype html>
<html lang="es">
  <head>
    <charset="utf-8">
    <title>Listado telefónico</title>
  </head>

  <body>
    <header>
      <h1>Listado telefónico</h1>
    </header>
    <table>
      <tr>
        <th>Nombre</th>
        <th>Apellido</th>
        <th>Teléfono</th>
      </tr>
      <!--SUSTITUCION-DINAMICA-->
      <tr>
        <td>[INDICADOR_DE_SUSTITUCION: NOMBRE]</td>
        <td>[INDICADOR_DE_SUSTITUCION: APELLIDO]</td>
        <td>[INDICADOR_DE_SUSTITUCION: TELÉFONO]</td>
      </tr>
      <!--SUSTITUCION-DINAMICA-->
    </table>
  </body>
</html>
```

## Plantilla de sustitución combinada:

```

<!doctype html>
<html lang="es">
  <head>
    <charset="utf-8">
    <title>[INDICADOR_DE_SUSTITUCION: TITULO]</title>
  </head>

  <body>
    <header>
      <h1>[INDICADOR_DE_SUSTITUCION: TITULO]</h1>
    </header>

    <article>
      [INDICADOR_DE_SUSTITUCION: CONTENIDO]
    </article>

    <h3>Listado telefónico</h3>
    <table>
      <tr>
        <th>Nombre</th>
        <th>Apellido</th>
        <th>Teléfono</th>
      </tr>
      <!--SUSTITUCION-DINAMICA-->
      <tr>
        <td>[INDICADOR_DE_SUSTITUCION: NOMBRE]</td>
        <td>[INDICADOR_DE_SUSTITUCION: APELLIDO]</td>
        <td>[INDICADOR_DE_SUSTITUCION: TELÉFONO]</td>
      </tr>
      <!--SUSTITUCION-DINAMICA-->
    </table>
  </body>
</html>

```

## Indicadores de sustitución en las GUI

Los indicadores de sustitución, son marcas de texto plano que se colocan en los archivos HTML para identificar las zonas de la plantilla, en las cuáles deben reemplazarse ciertos datos.

En **Python**, los identificadores de sustitución se expresan con **palabras concatenadas anteceditas del signo dólar (\$)**:

```

<!doctype html>
<html lang="es">
  <head>
    <charset="utf-8">
    <title>$titulo</title>
  </head>

  <body>
    <header>

```

```

        <h1>${titulo}</h1>
    </header>

    <article>
        $contenido
    </article>
</body>
</html>

```

Mientras que en **PHP**, no existe ninguna regla al respecto, quedando al libre albedrío del diseñador, la elección de estilo de los identificadores de sustitución. Un clásico en los identificadores de sustitución en PHP, es **encerrar las palabras identificadoras entre dos llaves, de apertura y cierre respectivamente {}**:

```

<!doctype html>
<html lang="es">
  <head>
    <charset="utf-8">
    <title>{titulo}</title>
  </head>

  <body>
    <header>
      <h1>{titulo}</h1>
    </header>

    <article>
      {contenido}
    </article>
  </body>
</html>

```

## Lógica de sustitución en MVC

La lógica de las vistas en MVC, es única y exclusiva para cada GUI en particular. No obstante, existe un procedimiento estándar, el cual consta de tres pasos:

1. Obtener el HTML (**GUI**);
2. Crear un **diccionario** con los identificadores de sustitución y los datos asociados a su reemplazo;
3. Realizar la **sustitución** de datos sobre la plantilla HTML e imprimirla en pantalla (es decir, mostrar el resultado al usuario).

### Obtener la GUI en PHP:

```
$plantilla = file_get_contents('/ruta/a/template.html');
```

### Obtener la GUI en Python:

```
with open('/ruta/a/template.html', 'r') as archivo:
```

```
plantilla = archivo.read()
```

### Crear un diccionario de sustituciones en PHP:

```
$diccionario = array(
    '{titulo}'=>'Página de muestra',
    '{contenido}'=>'Hola Mundo'
);
```

### Crear un diccionario de sustituciones en Python:

```
diccionario = dict(
    titulo='Página de prueba',
    contenido='Hola Mundo'
)
```

*Por favor, notar que en Python, el nombre de las claves del diccionario es el identificador de sustitución SIN el signo dólar, mientras que en PHP, las claves del diccionario deberán verse exactamente igual a los identificadores de sustitución utilizados en la plantilla HTML.*

### Sustitución de diccionarios en PHP:

```
$render = str_replace(array_keys($diccionario),
    array_values($diccionario), $plantilla);
```

### Sustitución de diccionarios en Python:

```
from string import Template
render = Template(plantilla).safe_substitute(diccionario)
```

## Sustituciones estáticas

Tanto en Python como en PHP, las sustituciones estáticas se realizan siguiendo los tres pasos estándar mencionados en el punto anterior: traer la plantilla, crear el diccionario y realizar la sustitución.

**Si los datos sustitutos debieran traerse desde una consulta a base de datos, primero se realizará la consulta SQL**, luego se almacenarán los datos retornados en variables y, finalmente, dichas variables se asignarán como valores de las claves del diccionario, sin más complejidad que ésta.

## Sustituciones dinámicas

En las sustituciones dinámicas, la mayor complejidad radica en la obtención del código HTML sobre el cual la sustitución debe realizarse.

La plantilla HTML se deberá obtener de forma estándar, mientras que de ella, antes de proceder con los dos pasos siguientes, se deberá recuperar sólo la fracción de código HTML sobre la cuál realizar los reemplazos. Dicha fracción de código, se obtendrá definiendo previamente, la siguiente **expresión regular**:

```
<!--NOMBRE-DE-LA-SUSTITUCION-DINAMICA-->(.\|\\n){1,}<!--NOMBRE-DE-LA-SUSTITUCION-DINAMICA-->
```

Dónde NOMBRE-DE-LA-SUSTITUCION-DINAMICA será el texto utilizado en el código HTML dentro del comentario identificador:

*Dado el siguiente identificador de sustitución dinámica:*

```
<!--TELEFONOS-->
... código html a iterar + indicadores de sustitución
<!--TELEFONOS-->
```

*La expresión regular debería verse como la siguiente:*

```
<!--TELEFONOS-->(.\|\\n){1,}<!--TELEFONOS-->
```

### Obtención del código HTML iterativo en PHP:

```
$regex = "/<!--TELEFONOS-->(.\|\\n){1,}<!--TELEFONOS-->/";
preg_match($regex, $plantilla, $matches); # $matches se define al vuelo
$match = $matches[0];
```

### Obtención del código HTML iterativo en Python:

```
import re
regex = re.compile("<!--TELEFONOS-->(.\|\\n){1,}<!--TELEFONOS-->")
match = regex.search(plantilla).group(0)
```

Una vez obtenido el match (fragmento de código HTML coincidente con la expresión regular), tanto el diccionario como las sustituciones, deberán realizarse de manera estándar pero dentro de un bucle, condicionado por la cantidad de registros obtenidos de la consulta SQL. La única salvedad, es que ambos pasos deberán realizarse en la misma estructura de control cíclica y, el resultado de cada sustitución, *sumarse* dentro de una misma variable. La sustitución, en este caso, se efectuará sobre el match y no sobre la plantilla.

### Sustitución iterativa en PHP:

```
$render = "";
foreach($registros as $array) {
    $diccionario = array(
        "{nombre}"=>$array[0],
        "{apellido}"=>$array[1],
        "{telefono}"=>$array[2]
    );
    $render .= str_replace(array_keys($diccionario),
```

```
        array_values($diccionario), $match);  
    }
```

### Sustitución iterativa en Python:

```
render = ""  
for tupla in registros:  
    diccionario = dict(  
        nombre=tupla[0],  
        apellido=tupla[1],  
        telefono=tupla[2]  
    )  
    render += Template(match).safe_substitute(diccionario)
```

Finalmente, deberá sustituirse el match por el render en la plantilla.

### Sustitución del match por el render en PHP:

```
$render_final = str_replace($match, $render, $plantilla);
```

### Sustitución del match por el render en Python:

```
render_final = plantilla.replace(match, render)
```

Si no se desea conservar los comentarios HTML identificadores de la sustitución dinámica (lo más recomendado), los mismos, se podrán reemplazar mediante la función `str_replace` en PHP y el método `replace` del objeto *string* de Python.

## Sustituciones combinadas

Cuando en una misma plantilla se necesite efectuar una sustitución estática y una o más sustituciones dinámicas, el proceso se hará paso a paso. Esto significa, que en primer lugar, se efectuará la sustitución estática. El resultado obtenido en ese render, deberá ser aquel sobre el cuál se realice el match para la sustitución dinámica. Es decir, que cuando se llegue al proceso de sustitución dinámica, **no se volverá a traer la plantilla**, sino que **se utilizará el código ya renderizado, obtenido en el proceso de sustitución estática**.

### *Tip by Commander in Chief:*

Evita utilizar servicios en línea para obtener un *hash* determinado. Estos servicios guardan la cadena ingresada asociándola al *hash* resultante y se utilizan para efectuar el proceso de ingeniería inversa sobre un *hash*, a fin de obtener la cadena original.

