

# Diseño de Aplicaciones para Internet (2018-2019)

## Guión de Prácticas 3:

### Plantillas (templates), Manejo de Sesiones y Persistencia Básica

S. Alonso, J.M. Guirao  
zerjioi@ugr.es, jmguirao@ugr.es

#### Resumen

En esta práctica avanzaremos en el uso de **Flask**: usaremos plantillas (templates) para generar cómodamente nuestras vistas, utilizaremos sesiones para gestionar la identificación de usuarios en la web y otros posibles datos de sesión de nuestra aplicación web y, por último, veremos como gestionar la persistencia usando un motor de bases de datos básica.

## 1. Plantillas (templates)

Cuando desarrollamos una aplicación web (por ejemplo con **Flask**) no es buena idea incluir el código HTML de las páginas dentro de nuestra aplicación Python. Utilizando plantillas (templates) conseguiremos simplificar mucho todas las tareas repetitivas, así como separar correctamente el aspecto de la aplicación (vista) de su lógica interna (controlador) [1].

Una biblioteca potente de templates para Python es **Jinja2** [2]. En esta práctica vamos a familiarizarnos con esta biblioteca. Para ello tendremos que construir varias páginas distintas.

Dichas páginas deberán mostrar un sitio web en el que haya (al menos) una cabecera, dos columnas y un pie de página. En la cabecera habrá una imagen de logo del sitio, el nombre del mismo, un subtítulo y un mini-formulario de **login**. En la columna izquierda habrá opciones (como por ejemplo menús), la columna de la derecha contendrá el cuerpo principal de la página y el pie contendrá información sobre el autor de la página y los derechos de la misma (licencia). En caso de no querer hacer el diseño de la página desde cero, podemos optar por descargar alguna plantilla web ya creada (por ejemplo de [3]) y adaptarla para poder presentar los contenidos dinámicos de la aplicación.

Debemos intentar seguir el paradigma *modelo, vista, controlador* (MVC) lo más fielmente posible, de tal manera que sea posible, por ejemplo, cambiar radicalmente el aspecto de la aplicación web modificando únicamente los templates. Los archivos de HTML estarán, por tanto, en un directorio separado.

## 2. Manejo de Sesiones

En toda aplicación web es necesario gestionar información que se mantenga entre las distintas páginas que visita el usuario. Para ello se hace uso de las sesiones [4].

En esta práctica adaptaremos nuestra página web para que el formulario de `login` anteriormente citado funcione correctamente y se muestre distinto contenido si el usuario está ya identificado o no (a esta altura la comprobación de un usuario único estará codificada “a pelo” en la aplicación, en el siguiente apartado guardaremos distintos usuarios usando persistencia). Por ejemplo, el formulario de login de la cabecera solo debe aparecerle a los usuarios no identificados, mientras que a los identificados debe mostrárseles el típico mensaje de “*Bienvenido xxxx*” y un enlace para hacer `logout`.

Además, implementaremos en nuestra aplicación (y utilizando exclusivamente sesiones) un “menu” que muestre -y permita acceder- a las últimas 3 páginas del sitio visitadas (lo podemos incluir en el menú de la izquierda).

## 3. Persistencia Sencilla: pickleshare

Para almacenar la información de nuestra aplicación podemos utilizar distintos mecanismos de persistencia. Cuando la aplicación sea suficientemente sencilla podemos utilizar algún esquema de almacenamiento local, como por ejemplo la biblioteca `pickleshare` [5].

En esta práctica usaremos dicha base de datos para almacenar la información de un usuario que se introduzca en un formulario de registro en nuestra aplicación web. Por tanto la identificación en la web (apartado anterior) tendrá que ser actualizado para permitir el acceso a cada usuario que se cree mediante este formulario. Crearemos una página nueva de nuestra aplicación que nos permita visualizar los datos del usuario (tendremos acceso a través del menú a esta página cuando estemos identificados en la web). Asimismo crearemos una nueva página que le permita al usuario cambiar los datos personales (volviendo a mandar el formulario, que aparecerá relleno con los datos anteriores).

Para instalar `pickleshare`:

```
> pip3 install pickleshare
```

## Referencias

- [1] Modelo Vista Controlador: [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)
- [2] Jinja2: <http://jinja.pocoo.org/>
- [3] Free Website Templates: <http://www.freewebsitetemplates.com/>
- [4] Sesiones, autenticación y control de acceso: <http://elpuig.xeill.net/Members/vcarceler/asix-m09/uf1/nf2/a5>
- [5] `pickleshare`: a small ‘shelve’ like datastore with concurrency support <https://pypi.org/project/pickleshare/>