

Práctica Unidades 4 y 5

Formularios, Cookies y POO

CFGS Desarrollo de Aplicaciones Web – Entornos de Desarrollo

Antes de comenzar

Recuerda:

- Usa las PEP-8 y las anotaciones de tipo.

En esta práctica tienes que completar una aplicación web que no es más que una tienda sencilla de productos que se pueden añadir a un carrito de la compra.

Para el desarrollo de la práctica usarás las clases que creaste en el último Ninja Training (**Product** y **ShoppingCart**), así que mueve el fichero **shopping_cart.py** a este proyecto.

Te propongo una serie de hitos para que el desarrollo de la práctica sea progresivo. Pero antes déjame que te explico cómo se van a almacenar los datos en esta aplicación web.

La base de datos

En esta práctica se van a almacenar los productos en una base de datos en un fichero. La base de datos se gestiona por medio de tres funciones que tienes en el fichero **database.py** que te proporciono yo. Estas son las tres funciones:

get_product: que recibe un uuid y devuelve el producto si existe un producto con ese uuid o None si no existe ese producto.

get_products: que no recibe nada y devuelve la lista de los productos que hay en la base de datos.

insert: que recibe un objeto producto y lo almacena en la base de datos (lo guarda en el fichero).

No te tienes que preocupar más que de cambiar el **import** que ves en la línea 3 para hacerla coincidir con tu fichero **shopping_cart.py**.

Para el desarrollo de la práctica no vas a necesitar ninguna otra operación sobre la base de datos. Como ves no se pueden eliminar productos. Si quieres borrar la base de datos para empezar de 0, simplemente elimina el fichero **src/database/data.json**.

Hito 1: empezando y añadir productos

Crea la estructura del proyecto que ves a continuación donde el enrutador estará en **src/server.py** y se llamará **app** (la función de entrada a la aplicación).

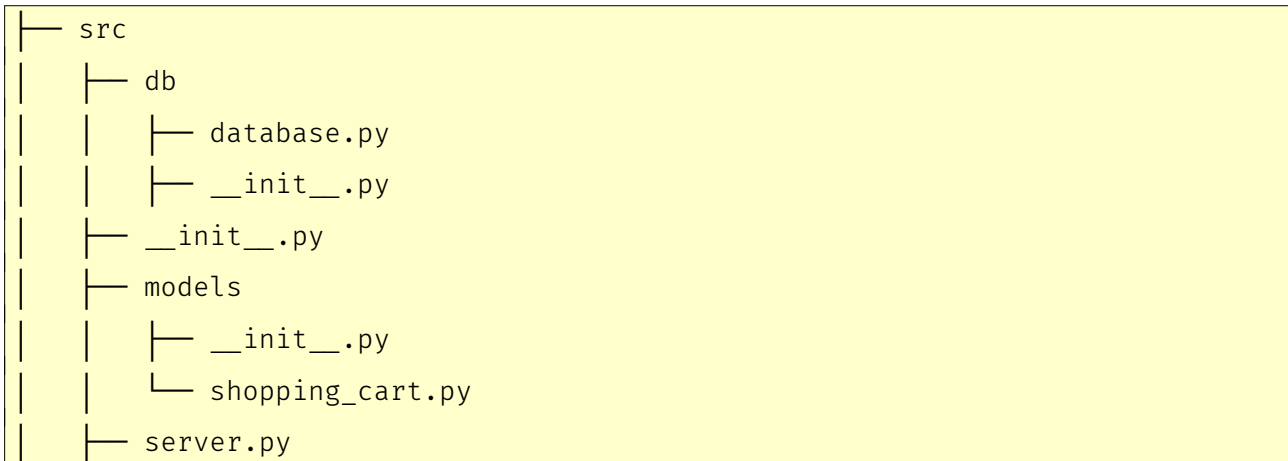
A partir de aquí puedes crear todas las carpetas y/o ficheros que estimes oportuno.

Fíjate que aparecen los ficheros:

- **shopping_cart.py** que tienes que sacar del Ninja Training,
- **database.py** que te lo doy yo y tienes que poner en **src/db**, y

- **server.py** que tienes que crear en **src**.

Todo lo demás es libre:



A partir de aquí, en este hito tienes que crear estas rutas:

- **/** para mostrar una página de bienvenida
- **/product/db/add** desde donde se podrá insertar un producto mediante un formulario como el que ves a continuación:

Añade un nuevo producto a la base de datos:

Nombre del producto:

Precio:

Descripción (opcional):

Añadir

Para la gestión del formulario debes tener en cuenta estos puntos:

- El nombre del producto es obligatorio (recuerda “strippear” lo que llegue).
- El precio es obligatorio (recuerda “strippear” lo que llegue) y, además, el precio tiene que ser un número (entero o real). Te muestro el código que usaría yo para comprobar si el string que nos llega en el campo precio es o no un número para poder, más tarde, convertirlo a **float**: (no es un código que puedas copiar y pegar en tu práctica, está fuera de contexto, pero sí es una guía que puedes usar para verificar si un string contiene un número entero o real:

```
if form["price"].replace(",", ".").replace(".", "").isnumeric():
    price: float = float(form["price"].replace(",", "."))
```

- La descripción es opcional pero no olvides “strippear” lo que llegue.

Los errores que se puedan cometer tienen que aparecer debajo de los campos del formulario.

Cuando llegue el formario correctamente cumplimentado se insertará el producto en la base de datos, llamando a la función **insert** del módulo **src/db/database.py**.

Resultados de aprendizaje y criterios de evaluación

- RA8. Genera páginas Web dinámicas analizando y utilizando tecnologías del servidor Web que añadan código al lenguaje de marcas.
 - d) Se han utilizado estas tecnologías para generar páginas Web que incluyan interacción con el usuario en forma de advertencias y peticiones de confirmación.
 - e) Se han utilizado estas tecnologías, para generar páginas Web que incluyan verificación de formularios.
 - f) Se han utilizado estas tecnologías para generar páginas web que incluyan modificación dinámica de su contenido y su estructura.
 - g) Se han aplicado estas tecnologías en la programación de aplicaciones Web.
- RA5. Desarrolla aplicaciones Web identificando y aplicando mecanismos para separar el código de presentación de la lógica de negocio.
 - a) Se han identificado las ventajas de separar la lógica de negocio de los aspectos de presentación de la aplicación.
 - f) Se han escrito aplicaciones Web con mantenimiento de estado y separación de la lógica de negocio.
 - g) Se han aplicado los principios de la programación orientada a objetos.
 - h) Se ha probado y documentado el código.