

Gestión de la Información en la Web

Curso 2017-18

Práctica MongoEngine

Fecha de entrega: jueves 14 de diciembre de 2017, 13:55h

Entrega de la práctica

La entrega de la práctica se realizará a través del Campus Virtual de la asignatura mediante un fichero `grXX_mongoengine.py` donde `XX` es el número de grupo.

Lenguaje de programación

Python 3.5 o superior.

Calificación

La definición del esquema suma hasta 9 puntos, y la creación de datos mínimos suma 1 punto. Además de la corrección se valorará la **concisión y claridad del código**, y la incorporación de **comentarios** explicativos.

Declaración de autoría e integridad

Todos los ficheros entregados contendrán una cabecera en la que se indique la asignatura, la práctica, el grupo y los autores. Esta cabecera también contendrá la siguiente declaración de integridad:

(Nombres completos de los autores) declaramos que esta solución es fruto exclusivamente de nuestro trabajo personal. No hemos sido ayudados por ninguna otra persona ni hemos obtenido la solución de fuentes externas, y tampoco hemos compartido nuestra solución con nadie. Declaramos además que no hemos realizado de manera deshonesto ninguna otra actividad que pueda mejorar nuestros resultados ni perjudicar los resultados de los demás.

No se corregirá ningún fichero que no venga acompañado de dicha cabecera.

En esta práctica definiremos y usaremos el esquema MongoEngine necesario para contactar con una base de datos MongoDB para una tienda *on-line*. Esta base de datos se debe llamar **giw_mongoengine** y debe soportar los siguientes *conceptos*:

Usuario

- DNI (obligatorio, único)
- Nombre (obligatorio)
- Primer apellidos (obligatorio)
- Segundo apellido (opcional)
- Fecha de nacimiento (obligatorio, formato 'AAAA-MM-DD')
- Fecha de los últimos 10 accesos al sistema (opcional, las fechas tienen el formato 'AAAA,MM,DD,HH,MM,SS,NNNNNN')
- Lista de **tarjetas de crédito** (opcional)
- Lista de referencias a **pedidos** (opcional)

Tarjeta de crédito

- Nombre del propietario (obligatorio)
- Número de la tarjeta (obligatorio, 16 dígitos)
- Mes de caducidad (obligatorio, 2 dígitos)
- Año de caducidad (obligatorio, 2 dígitos)
- Código de verificación de tarjeta o *CVV* (obligatorio, 3 dígitos)

Pedido

- Precio total del pedido (obligatorio)
- Fecha del pedido (obligatorio, 'AAAA,MM,DD,HH,MM,SS,NNNNNN')
- Lista de **líneas del pedido** (obligatorio)

Línea de pedido

- Cantidad de productos comprados (obligatorio)
- Precio de un producto (obligatorio)
- Nombre del producto (obligatorio)
- Precio total de la línea (obligatorio)
- Referencia al **producto** (obligatorio)

Producto

- Código de barras (obligatorio, único, formato EAN-13)
- Nombre (obligatorio)
- Categoría principal (obligatorio, número natural)
- Lista de categorías secundarias (opcional, números naturales)

1. Definición del esquema [9pt]

Crea en un fichero `grXX.mongoengine.py` (donde `XX` es el número de grupo) las clases necesarias para representar el esquema anteriormente detallado en MongoEngine. Además de las comprobaciones naturales (longitudes, valores mínimos, etc.) el esquema debe realizar las siguientes validaciones adicionales:

1. El formato del DNI de los usuarios es correcto, incluyendo el dígito de control (<http://www.interior.gob.es/web/servicios-al-ciudadano/dni/calculo-del-digito-de-control-del-nif-nie>).
2. El precio total de un pedido es exactamente la suma de los precios de todas sus líneas.
3. El precio total de una línea de pedido está correctamente calculado en base al precio de un producto y la cantidad de productos comprados.
4. El nombre del producto de una línea de pedido es el mismo que el del producto al que apunta la referencia.
5. El código de barras EAN-13 de un producto está bien formado, incluyendo su dígito de control (https://en.wikipedia.org/wiki/International_Article_Number.)
6. Si un producto tiene categorías secundarias, su categoría principal debe aparecer en el primer lugar de esa lista.
7. Cuando un pedido se elimina, este debe desaparecer de la lista de pedidos del usuario que lo realizó.

2. Inserción de datos [1pt]

Implementar una función `insertar()` que crea los objetos Python necesarios y los utiliza para insertar en MongoDB (como mínimo) 2 usuarios, cada uno de ellos con al menos 2 pedidos diferentes de más de una línea. Al menos un usuario debe tener 2 o más tarjetas de crédito.