



PROJECTE WEB

1st Project Deliverable



Universitat de Lleida





INDEX

1. Introducció.....	3
2. Anàlisis fitxers més importants.....	4
3. Reflexions generals.....	6
4. Comandes importants projecte.....	7
5. Link pràctica “GitHub”	7



1. Introducció

En aquesta primera entrega, hem començat a definir una aplicació web amb l'ajuda del programa “PyCharm”, basada en l'àmbit de compra en un supermercat que serà d'ajuda per als clients del supermercat tractats a la nostra aplicació web, les cadenes administradores dels supermercats podran afegir, eliminar, retocar productes i editar paràmetres de preus que es visualitzaran en cadascun dels productes, i de cara als clients, poder extreure informació sobre els productes d'un article i poder comprar-ne i fer una comanda d'algun producte en especial.

2. Anàlisi fitxers més importants

La mare de la nostra aplicació web es al moment que creem el projecte, on s'han creat els quatre fitxers per defecte, ja que un dels arxius més importants és “urls.py” ubicat “proyectoshop/shop/**shop**” on s'especifica la ruta o la ubicació per anar a tots els recursos/elements de la nostra aplicació, sense això no ens funcionaria absolutament res de la nostra aplicació bàsica (app) que hem creat dintre de la carpeta principal del projecte “proyectoshop/shop/**shoponline**”. A continuació mostrem una part del contingut més irrellevant del fitxer “urls.py”.

```
from django.contrib import admin
from django.urls import path

urlpatterns = [
    path('admin/', admin.site.urls),
]
```


Imatge 1: Part del codi del fitxer “urls.py”

A continuació mostrem el contingut del fitxer “settings.py” on s'especifica totes les “apps” que s'han instal·lat al nostre sistema (**shoponline**).

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'shoponline',
]
```

Imatge 2: Part del codi del fitxer “settings.py”



Dintre de la nostra “app” en qüestió, que s’anomena “shoponline”, és on hem desenvolupat tota la feina de la nostra aplicació web bàsica, on tenim per una banda l’arxiu “models.py” que és on hem definit les nostres classes en l’àmbit de base de dades on cada classe creada serà una nova taula, i cada variable introduïda dintre d’aquestes classes, una nova columna on podrem afegir-hi atributs (llenguatge “python” basat amb “sql”). Per veure una mica l’aspecte del nostre fitxer “models.py”, mostrarem un petit exemple de classes.

```
class Categoria(models.Model):
    nombre = models.CharField(max_length=30)
    descripcion = models.CharField(max_length=200)

    def __str__(self):
        return 'Categoria : ' + self.nombre

class Productos(models.Model):
    nombre = models.CharField(max_length=30)
    precio = models.FloatField()
    categoria = models.ForeignKey('Categoria' , on_delete=models.SET_NULL , null=True)
```

***Imatge 3:** Part del codi del fitxer “models.py”*

A continuació us mostrem un petit exemple de “admin.py”.

```
admin.site.register(Usuario)
admin.site.register(Productos)
admin.site.register(Categoria)
admin.site.register(Envio)
admin.site.register(Orden)
```

***Imatge 4:** Part del codi del fitxer “admin.py”*

Després de tot això, hem desplegat la interfície de Django per a poder afegir, eliminar i modificar instàncies, i a més a més hem afegit l’autenticació d’usuari per poder entrar a la interfície.

Finalment explicarem l’última part de la nostra pràctica, la part on despleguem la nostra aplicació web bàsica. És la part on l’usuari veu tot el contingut de l’aplicació per mitjà del seu navegador, en aquest cas hem fet servir **Heroku**.

A continuació mostrarem la pàgina web de Heroku desplegada malgrat encara no hem introduït ni implementat la part de HTML ni de CSS que no es demana amb aquesta primera entrega.

Heroku | Welcome to your new app!

Refer to the [documentation](#) if you need help deploying.

Imatge 5: Pàgina Heroku.

3. Reflexions generals:

Ha set una pràctica força entretinguda ja que el poder programar alguna cosa i després que tingui forma visualitzant-ho al navegador web es una forma diferent de fer les coses amb el que estem acostumats.

Respecte al projecte de “github” els “commits” ens ha costat una mica acostumar-nos al fet de tenir que pujar contínuament tot el projecte o els fitxers que s’han modificat.

Ens ha costat entendre més que qualsevol altra pràctica de programació, primerament per situar-nos i saber la jerarquia de fitxers i les seves funcionalitats i després per entendre el tipus de llenguatge i per a que serveix cada funció de les “apps”, per això hem anat treballant cada dia una mica i fins i tot hem quedat amb el Roberto García Gonzalez en una ocasió per un error que ens ha costat resoldre que finalment ha set una tontería i una amb en Carles Mateu Piñol per anar a solucionar un petit detall que ens feia encallar en l’elaboració d’aquesta primera entrega.

Finalment la part del Docker-Compose, part que no hem pogut implementar perquè ens surten molts errors que no hem sapigut resoldre, i també perquè no estem familiaritzats amb aquesta eina, però, seguirem treballant per poder implementar el Docker per tal de seguir amb la següent entrega.



4. Comandes importants projecte:

- a) **Instal·lar Django:** \$ pipenv shell
 \$ pipenv instalar Django
- b) **Iniciar interfície Django:** \$ django-admin startproject proyectoshop
- c) **Per afegir els models, editem el fitxer “models.py” guardamos i fem servir les comandes següents:** \$ python manage.py makemigrations shoponline
 \$ python manage.py migrate shoponline
- d) **Per llençar el Heroku:** \$ heroku open

5. Links pràctica:

- a) GitHub: <https://github.com/duberdz/proyectoshop>
- b) Heroku app: <https://proyectoweb-shop.herokuapp.com/>
- c) Usuari i contrasenya: [“Usuari: **admin**, Contrasenya: **admin**”](#)