1. Tareas 6-13 mayo

Tareas a hacer

1. Peticiones web para un listado de los laboratorios (api rest CRUD)
2. Gestor de Base de datos -> SQLlite. Perfecto para pocos datos. Otra opción PostgreSQL de django
3. Cargar fichero de reservar

- [ ] 1.
- [x] 2.
- [ ] 3

1. Creación web en Django

Lo primero de todo, antes de poder ejecutar el servidor hay que ir a la ruta:
`/Documentos/git/Apoyo-Docencia/codigo-proyecto/mysite`
Dentro de esta ruta ejecutar el siguiente comando:
`python3 manage.py runserver`
Con esto podemos ver el servidor está en funcionamiento cómo se puede ver
`Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 06, 2022 - 16:17:21
Django version 4.0.4, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
`

Si le damos click a la url que nos ofrece, nos llevará a la página por defecto creada por Django.
IMPORTANTE, esta url es solo para developoing no para producción.

1.1. Creacíón de modelos

Para poder crear las apps que queramos, tendremos que irnos al terminal primero, en la misma ruta que el paso anterior e intrudicr el siguiente comando:
`python3 manage.py startapp nombre_de_la_app`
Dentro de `nombre_de_la_app/views.py` se puede ver el contenido de la vista y ahí será dónde la modificaremos. Para poder conectarla con el *main* tendremos que dirigirnos a `nombre_de_la_app/urls.py` y ahí introducir el siguiente código:

```
from django.urls import path

from . import views

urlpatterns = [
    path('', views.index, name='index'),
]
```

Y para la parte del *main* hacer algo parecido de la ruta: `mysite/urls.py`

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

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

Para acceder a la vista de nuestra app tendremos que ir a la siguiente url:
http://localhost:8000/nombre_de_la_app/

###Creación de clases
Para la creación de clases, nos vamos a nuestra app creada, al fichero `modules.py` y ahí podremos crear las distintas clases que queramos para esa app. Un ejemplo de ello sería el que podemos ver aqui:

```
from django.db import models


class Classroom(models.Model):
    specification = models.CharField(max_length=100)
    location = models.CharField(max_length=150)
    num_pc = models.BigIntegerField(default=0)
    s_o = models.CharField(max_length=20)
    num_class = models.CharField(max_length=10)
    capacity = models.CharField(max_length=20)
    specialization = models.CharField(max_length=100)
```

Para el tipo de atributo mirar la siguiente url. Para poder activar esta clase, hay que irse a `mysite/settings.py` y añadir nuestra clase

```
INSTALLED_APPS = [
    'nombre_de_la_clase.apps.Nombre_de_la_claseConfig',
    'django.contrib.admin',
    'django.contrib.auth',
```

```
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

¡Ojo con las mayúsculas!
Ahora tendremos que decirle a Django que hemos hecho unos cambios a mis modales, por lo tanto hay que ejecutar el siguiente código `python3 manage.py makemigrations nombre_de_la_clase` para crear *migrations* para esos cambios.
Para poder tener el esquema sql de nuestro modal, lo que tendremos que hacer es ejecutar el siguiente comando `python3 manage.py sqlmigrate nombre_de_la_clase 0001` arrojándonos este resultado:

```
BEGIN;
--
-- Create model Classroom
--
CREATE TABLE "classroom_classroom" (
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "specification" varchar(100) NOT NULL,
    "location" varchar(150) NOT NULL,
    "num_pc" bigint NOT NULL,
    "s_o" varchar(20) NOT NULL,
    "num_class" varchar(10) NOT NULL,
    "capacity" varchar(20) NOT NULL,
    "specialization" varchar(100) NOT NULL
);
COMMIT;
```

Para ahora crear estas tablas dentro de la BD ejecutamos el siguiente comando `python3 manage.py migrate` y nos dirá que ha aplicado el *migrate* de la tabla que hemos creado.

1.2. Modificación de la interfaz