
PUCP

Desarrollo Web con Python

CETAM - PUCP

Sesión 5:

- Django Framework
  - HTML Estático
  - HTML Dinámico
  - Formularios
- Django - Base de datos
  - SQLite
  - PostgreSQL
  - MySQL
  - Conexión a una base de datos

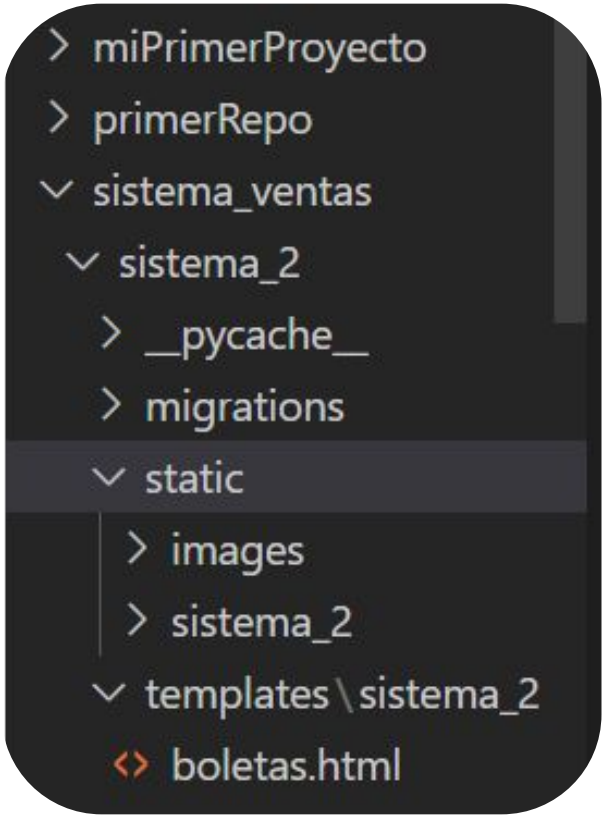
Django Framework

Desarrollo Web con Python - Sesión 5

Directorios Templates y Static

En estos directorios se almacenan las vistas y estilos de la aplicación web a desarrollar. Cabe resaltar que los HTML programados pueden tener porciones de código en python, es por esto que debe renderizarse a través de las funciones descritas en views.py. En el caso del directorio static, los archivos a guardar son los estilos y las imágenes del proyecto.

Directorios Templates y Static



HTML Estático

```
def index(request):
    return render(request,'django_ejemplo/index.html')
```

HTML Dinámico

```
def index(request):
    return render(request,'django_ejemplo/index.html',{
        'nombre': 'Alexander',
        'participantes': participantes
    })
```

HTML Dinámico

```
def index(request,ind):
    print(ind)
    mensaje = 'Se ha recibido el identificador ' + str(ind)
    return render(request,'django_ejemplo/index.html',{
        'mensaje' : mensaje
    })
```

HTML Formularios

```
def index(request,ind):
    if request.method == 'POST':
        nombre = request.POST.get('nombre')
        apellido = request.POST.get('apellido')
        ocupacion = request.POST.get('ocupacion')
        mensaje = 'Participante guardado'
    return render(request,'django_ejemplo/index.html',{
        'mensaje': mensaje
    })
```

Peticiones GET


```
def index(request,ind):
    if request.method == 'GET':
        nombre = request.GET.get('colour')
        apellido = request.GET.get('filtrar')
        mensaje = 'Participante guardado'
    return render(request,'django_ejemplo/index.html',{
        'mensaje': mensaje
    })
```

Django - Base de datos

Desarrollo Web con Python - Sesión 5

Bases de datos

Las bases de datos es el espacio en donde la información es almacenada. Esta configuración se realiza en el archivo settings.py del proyecto principal. Por defecto el proyecto django se crea con una base de datos SQLite.

Bases de datos

Django ofrece soporte nativo para una variedad de bases de datos, entre las principales:

- SQLite
- MySQL
- PostgreSQL

SQLite

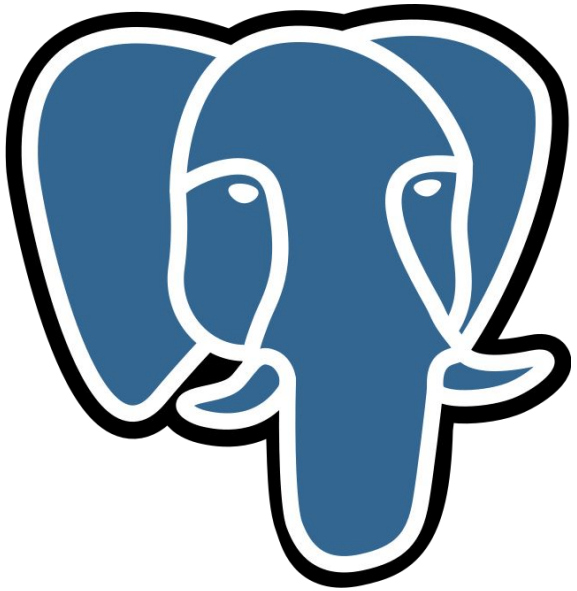
Es una base de datos ligera y portable, es la base de datos por defecto de django. La configuración se puede observar en el archivo settings del proyecto. Se utiliza en aplicaciones con poca cantidad de datos ya que no se requiere de un servidor

MySQL

Es la base de datos más popular dentro de todas las que son relacionales. Al igual que postgreSQL y a diferencia de SQLite se requiere el uso de un servidor para su funcionamiento. Este tipo de base de datos pueden ser hospedadas en servidores o en sistemas en la nube. Es ideal para manejar una cantidad de datos moderada y requiere menos recursos que PostgreSQL

PostgreSQL

Es la base de datos más avanzada de todas las mencionadas ya que permite manejar diferentes estructuras de datos, desde valores numéricos hasta arreglos, e incluso permite establecer nuevos tipos de datos. Es ideal para manejar grandes volúmenes de datos del orden de varios terabytes.



Sentencias SQL

Las sentencias SQL más comunes son:

- CREATE TABLE <nombre_tabla>(
  id INTEGER PRIMARY KEY,
  nombre TEXT NOT NULL
)
- SELECT * FROM <nombre_tabla>
- INSERT INTO <nombre_tabla> VALUES (2,'Alexander')
- SELECT * FROM <nombre_tabla> WHERE <condicion>
- DELETE FROM <nombre_tabla> WHERE <condicion>

Django ORM

Desde el shell de django podemos acceder a la base de datos y configurar la información almacenada.

python manage.py shell
<nombre_clase>.objects.all()

Conectando a una base de datos

```
"""
#Base de datos de prueba
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'devMetal',
        'USER': 'devMetal',
        'PASSWORD': 'devMetal',
        'HOST': 'database-2.cvwnf9ht2gui.us-east-1.rds.amazonaws.com',
        'PORT': '2233',
        #'ENGINE': 'django.db.backends.sqlite3',
        #'NAME': BASE_DIR / 'db.sqlite3',
    }
}

"""
#Base de datos de produccion
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'metalprotec',
        'USER': 'metalprotec',
        'PASSWORD': 'metalprotec',
        'HOST': 'database-1.cvwnf9ht2gui.us-east-1.rds.amazonaws.com',
        'PORT': '54321',
        #'ENGINE': 'django.db.backends.sqlite3',
        #'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

Gracias