

Universidad Mariano Gálvez de Guatemala.
Facultad de Ingeniería de Información y Ciencias de la Computación.

Curso: 029-B Sistemas Operativos I
Catedrático: Juan Fernando Contreras Giron



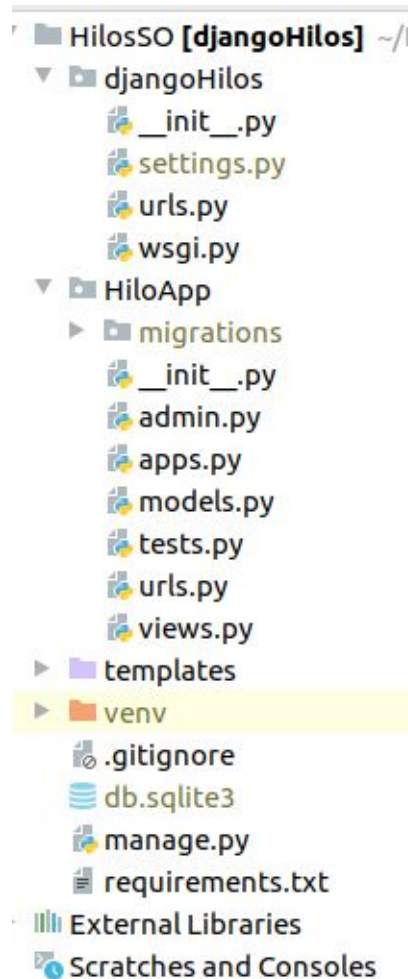
Manual Técnico Hilos

Nombre:

Cristhian Alexis Recinos 1990-16-11653
Edy Santiago Saput Xoyón 1990-17-13701
Juan Carlos Catú Similox 1990-17-18018
Hans Levi Giron De León 1990-16-14934
Jaime David Bal Chex 1990-17-7652
Fecha: 02/11/2019

El programa realizado para ejemplificar “Hilos”, esta echo en codigo python, especificamente en el framework Django, Los requisitos previos para correr el programa se necesita tener su entrono virtual corriendo, en el siguiente link le muestra como crearlo, https://developer.mozilla.org/es/docs/Learn/Server-side/Django/development_environment, he instalar las dependencias que se encuentran en: “requirements.txt” en la carpeta del proyecto.

Teniendo todo lo anterior puedo correr el programa, el programa se estructura de la siguiente manera:



← La carpeta djangoHilos: Esta carpeta contiene la configuración general del proyecto.

← la carpeta HiloApp: contiene la configuración de la aplicación del proyecto es decir: el codigo como tal, específicamente en el archivo “views.py”

The screenshot shows a file explorer window titled 'HilosSO [djangoHilos] ~/l'. The tree structure is as follows:

- djangoHilos
 - __init__.py
 - settings.py
 - urls.py
 - wsgi.py
- HiloApp
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
 - templates
 - venv
- .gitignore
- db.sqlite3
- manage.py
- requirements.txt
- External Libraries
- Scratches and Consoles

HiloApp/views.py:

```
views.py | djangoHilos/urls.py | HiloApp/urls.py
1 from django.shortcuts import render
2 import time
3 import threading
4
5 # Create your views here.
6 datos = {}
7
8 class hilo:
9     # metodo aceptar personas -----Hilo1
10     def personass(personas):
11         for person in personas:
12             mensajeP = 'Hilo No.1 - Hola querido ' + person + ' Como estas?'
13             print(mensajeP)
14             datos.append(mensajeP)
15             time.sleep(0.5)
16
17     # metodo asignarle un id lista----- Hilo2
18     def asignarleId(personas):
19         i = 1
20         for person in personas:
21             mensajeId = "Hilo No.2 - Hola! {}, tu id es {}".format(person, i)
22             print(mensajeId)
23             datos.append(mensajeId)
24             i += 1
25             time.sleep(0.5)
```

En las Líneas 1,2,3. importamos las librerías a usar, render, para renderizar a nuestra página “index.html”, time: libreria python que proporciona muchas formas de representar el tiempo en el código. threading: para la utilización de hilos.

datos = {}: esta es la forma de declarar diccionarios en python.

Nosotros creamos una clase llamada “hilo” para definir dentro de esa clase nuestras

funciones.

La función personass recibe como parámetro del diccionario lista (más adelante se explicara de esto), y recorrerá cada elemento de dicha lista he imprimirá en consola he imprimirá en consola y lo guardará en el diccionario datos.
el método time.sleep() es el tiempo que tardará cada iteración.

La función asignarleId(personas) ← esta funciona recibe el mismo diccionario llamado lista pero le asignar un id a cada persona. tiene el mismo tiempo que el metodo personass.

```
27 # metodo asignar Departamento -----Hilo3
28 def asignarDepartamento(personas, departamentos):
29     i = 0
30     for departamento in departamentos:
31         mensajeC = 'Hilo No.3 - Ciudad para {}, es {}'.format(personas[i], departamento)
32         print(mensajeC)
33         datos.append(mensajeC)
34         i += 1
35         time.sleep(0.5)
36
37
38 def index(request):
39     t = time.time()
40     datos.clear() # limpiamos los datos
41     lista = ['Alexis', 'Juan', 'Santiago', 'Hans', 'David']
42     departamentos = ['Chimaltenango', 'Comalapa', 'Chimaltenango', 'Chimaltenango', 'Comalapa']
43     # -----Creando Hilos Para las Personas, sus Id's y departamento-----
44     t1 = threading.Thread(target=hola.personass, args=(lista,))
45     t2 = threading.Thread(target=hola.asignarleId, args=(lista,))
46     t3 = threading.Thread(target=hola.asignarDepartamento, args=(lista, departamentos))
47     # -----Iniciamos Los Hilos-----
48     t1.start()
49     t2.start()
50     t3.start()
51     t1.join()
52     t2.join()
53     t3.join()
```

El método
asignarDepartamento(personas,
departamentos) ← recibe dos
parámetros los cuales son los
diccionarios lista y departamentos.

El método index ← este método
recibe una solicitud del archivo
urls.py y lo ejecuta. el método time
es instanciado y lo guarda en una
variable “t”, datos.clear() ← limpia

o borra todos los datos de la variables datos={}, lista: se declara el diccionario de los nombres a procesar, departamentos: se declara el diccionario de los departamentos a procesar.

el método threading.Thread(target=hola.personass, args=(lista,)) ← sirve para usar los hilos en python, y recibe dos parámetros la función donde va el hilo y mandamos la el diccionario lista y guardamos este método en una variable llamada “t1”, así sucesivamente para t2 y t3.

los métodos t1.start(), t2.start(), t3.start() ← Iniciamos los hilos.

con los métodos `t1.join()`, `t2.join()`, `t3.join()` ← Una vez terminado los procesos los hilos termina.

```
54     tiempo = str(time.time() - t)
55     print("El tiempo tardado " + str(time.time() - t))
56     return render(request, "index.html", {'persona': datos, 'tiemp': tiempo})
57
```

creamos una variable llama "tiempo" la cual capturamos el tiempo transcurrido de todos los hilos y lo imprimimos en consola.

El método `return render()` es para renderizar al html el diccionario `datos`, y la variable `tiempo`.

Archivo

HiloApp/urls.py

```
1 from django.urls import path
2 from .views import *
3
4 urlpatterns = [
5     path('', index, name="index"),
6 ]
```

Al momento de que el usuario entre a la pagina, llama al archivo `urls.py` y este llama al metodo `index` que se encuentra en `HiloApp/views.py`.

Archivo

templates:

acá es donde está nuestro Frontend: creamos una estructura de html común, en las clases de los divs son propiedades de la herramienta bootstrap. he usamos un `for` que nos permite el framework Django la cual es: `{% For per in persona %}` y colocamos cada elemento `{{ per }}` en una etiqueta ``.

mostramos el tiempo transcurrido `{{ tiemp }}` en una etiqueta `<h2></h2>`

```
17 {#-----bloque principal-----#}
18 {% block content %}
19     <div class="container-fluid">
20         <div class="card">
21             <div class="card-header text-center">
22                 <div class="row">
23                     <div class="col-10">
24                         <h2>Hilos en Python With Django</h2>
25                     </div>
26                     <div class="col">
27                         <a class="btn btn-primary" href="/" role="button">Empezar de Nuevo</a>
28                     </div>
29                 </div>
30             </div>
31             <div class="card-body">
32                 {% for per in persona %}
33                 <ul class="list-group list-group-flush">
34                     <li class="list-group-item">
35                         {{ per }}
36                     </li>
37                 </ul>
38                 {% endfor %}
39             </div>
40             <div class="card-footer text-muted text-center">
41                 <h2>Tiempo Transcurrido {{ tiemp }}</h2>
42             </div>
43         </div>
44     </div>
45 {% endblock %}
```