

GUÍA DE LABORATORIO 5

INFORMACIÓN BÁSICA					
ASIGNATURA:	Programación Web 2				
TEMA:	Django				
NÚMERO DE PÁCTICA:	05	AÑO LECTIVO:	2023 A	NRO. SEMESTRE:	III
FECHA DE INICIO:	29-May-2023	FACHA FIN:	05-Junio-2023	DURACIÓN:	4 horas
AUTOR: Jhamil Yeyder Turpo Añasco					
DOCENTE: Aníbal Sardon Paniagua					

DJANGO

1. COMPETENCIAS DEL CURSO

- Diseña responsablemente aplicaciones web, sus componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad.
- Construye responsablemente soluciones con tecnología web siguiendo un proceso adecuado llevando a cabo las pruebas ajustada a los recursos disponibles del cliente.
- Aplica de forma flexible técnicas, métodos, principios, normas, estándares y herramientas del desarrollo web necesarias para la construcción de aplicaciones web e implementación de estos sistemas en una organización.

2. EJERCICIOS PROPUESTOS

Listings

- | | | |
|----|---|---|
| 1. | Estructura de los repositorios y archivos | 2 |
| 2. | Ajustes que se toman en cuenta en settings | 3 |
| 3. | Código de urls del proyecto | 3 |
| 4. | Código de models (campos creados) | 4 |
| 5. | Código de urls de la aplicación catalog (redireccionamientos) | 5 |
| 6. | Código de vistas (administración de contenido) | 6 |

- Cree la aplicación Library paso a paso desde la siguiente url:
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website

Listing 1: Estructura de los repositorios y archivos

```
1  Library
2  |   db.sqlite3
3  |   manage.py
4  |
5  +---Apps
6  |   \---catalog
7  |       |   admin.py
8  |       |   apps.py
9  |       |   models.py
10 |       |   tests.py
11 |       |   urls.py
12 |       |   views.py
13 |       |   __init__.py
14 |       |
15 |       +---migrations
16 |       +---static
17 |       |   \---css
18 |       |       styles.css
19 |       |
20 |       +---Templates
21 |       |   |   base.html
22 |       |   |   index.html
23 |       |   |
24 |       |   +---catalog
25 |       |       |   author_detail.html
26 |       |       |   author_list.html
27 |       |       |   book_detail.html
28 |       |       |   book_list.html
29 |       |       |
30 |       |       \---registration
31 |       |           |   logged_out.html
32 |       |           |   login.html
33 |       |           |   password_reset_complete.html
34 |       |           |   password_reset_confirm.html
35 |       |           |   password_reset_done.html
36 |       |           |   password_reset_email.html
37 |       |           |   password_reset_form.html
38 |       |
39 |       \---__pycache__
40 |
41 +---images
42 |   |   django1.png
43 |   |   django2(esqueleto).png
44 |   |   django3(admin).png
45 |   |   django4(administracion).png
46 |   |   django5(viewpersonal).png
47 |   |   django6(libros).png
48 |   |   django7(paginacion).png
49 |   |   django8(autores).png
50 |   |   django9(autor).png
51 |
52 \---Library
53 |   asgi.py
54 |   settings.py
55 |   urls.py
56 |   wsgi.py
57 |   __init__.py
58 |
59 \---__pycache__
60
```

Los archivos más importantes que se verán a continuación son:

- En Library:

Listing 2: Ajustes que se toman en cuenta en settings

```
1 import os
2 from pathlib import Path
3 INSTALLED_APPS = [
4     'django.contrib.admin',
5     'django.contrib.auth',
6     'django.contrib.contenttypes',
7     'django.contrib.sessions',
8     'django.contrib.messages',
9     'django.contrib.staticfiles',
10    'Apps.catalog.apps.CatalogConfig', #aplicación library
11 ]
12 TEMPLATES = [
13     {
14         'BACKEND': 'django.template.backends.django.DjangoTemplates',
15         'DIRS': [os.path.join(BASE_DIR, 'Apps/catalog/Templates')],
16         'APP_DIRS': True,
17         'OPTIONS': {
18             'context_processors': [
19                 'django.template.context_processors.debug',
20                 'django.template.context_processors.request',
21                 'django.contrib.auth.context_processors.auth',
22                 'django.contrib.messages.context_processors.messages',
23             ],
24         },
25     ],
26 ]
27 LANGUAGE_CODE = 'es'
28
29 TIME_ZONE = 'America/Lima'
30 # actualización de cookies en cada session
31 SESSION_SAVE_EVERY_REQUEST = True
32
33 # redirección a la página de inicio
34 LOGIN_REDIRECT_URL = '/'
35 # para reestablecer contraseña
36 EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
```

Listing 3: Código de urls del proyecto

```
1 from django.contrib import admin
2 from django.urls import path, include
3 from django.views.generic import RedirectView
4
5 urlpatterns = [
6     path('accounts/', include('django.contrib.auth.urls')), # (for login, logout,
7     path('admin/', admin.site.urls),
8     path('catalog/', include('Apps.catalog.urls')),
9     path('', RedirectView.as_view(url='catalog/', permanent=True)),
10 ]
11 #archivos estaticos
12 from django.conf import settings
13 from django.conf.urls.static import static
14
15 urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

- En la App catalog:

Listing 4: Código de models (campos creados)

```
1 from django.db import models
2 from django.urls import reverse
3 import uuid # Required for unique book instances
4 from datetime import date
5 from django.contrib.auth.models import User
6
7 # Create your models here.
8
9 class Genre(models.Model):
10     """Model representing a book genre."""
11     name = models.CharField(max_length=200, help_text='Enter a book genre (e.g. Science Fiction)')
12
13     def __str__(self):
14         """String for representing the Model object."""
15         return self.name
16
17 # adicional :)
18 class Language(models.Model):
19     """Model representing a Language (e.g. English, French, Japanese, etc.)"""
20     name = models.CharField(max_length=200,
21                             help_text="Enter the book's natural language (e.g. English, French, Japanese etc.)")
22
23     def __str__(self):
24         """String for representing the Model object (in Admin site etc.)"""
25         return self.name
26
27
28 class Book(models.Model):
29     title = models.CharField(max_length=200)
30
31     # Foreign Key porque solo tendra un autor, pero un autor puede tener muchos libros
32     author = models.ForeignKey('Author', on_delete=models.SET_NULL, null=True)
33
34     summary = models.TextField(max_length=1000, help_text='Enter a brief description of the book')
35     isbn = models.CharField('ISBN', max_length=13, unique=True,
36                             help_text='13 Character <a href="https://www.isbn-international.org/content/what-isbn">ISBN number</a>')
37
38     # ManyToManyField tiene relacion de muchos a muchos
39     genre = models.ManyToManyField(Genre, help_text='Select a genre for this book')
40
41     language = models.ForeignKey('Language', on_delete=models.SET_NULL, null=True)
42
43     class Meta:
44         ordering = ['title', 'author']
45
46     def display_genre(self):
47         return ', '.join([genre.name for genre in self.genre.all()[:3]])
48
49     display_genre.short_description = 'Genre'
50
51     def __str__(self):
52         """String for representing the Model object."""
53         return self.title
54
55     def get_absolute_url(self):
56         """Returns the URL to access a detail record for this book."""
57         return reverse('book-detail', args=[str(self.id)])
```

```

58
59 class BookInstance(models.Model):
60     """Model representing a specific copy of a book (i.e. that can be borrowed from the
        library)."""
61     id = models.UUIDField(primary_key=True, default=uuid.uuid4, help_text='Unique ID
        for this particular book across whole library')
62     book = models.ForeignKey('Book', on_delete=models.RESTRICT, null=True)
63     imprint = models.CharField(max_length=200)
64     due_back = models.DateField(null=True, blank=True) # fecha de disponibilidad del
        libro
65     borrower = models.ForeignKey(User, on_delete=models.SET_NULL, null=True, blank=True
        )
66
67     @property
68     def is_overdue(self):
69         """Determines if the book is overdue based on due date and current date."""
70         return bool(self.due_back and date.today() > self.due_back)
71
72     LOAN_STATUS = (
73         ('m', 'Maintenance'),
74         ('o', 'On loan'),
75         ('a', 'Available'),
76         ('r', 'Reserved'),
77     )
78
79     status = models.CharField(
80         max_length=1,
81         choices=LOAN_STATUS,
82         blank=True,
83         default='m',
84         help_text='Book availability',
85     )
86
87     class Meta:
88         ordering = ['due_back']
89         permissions = (('can_mark_returned', 'Set book as returned'),)
90
91     def __str__(self):
92         """String for representing the Model object."""
93         return f'{self.id} ({self.book.title})'
94
95 class Author(models.Model):
96     """Model representing an author."""
97     first_name = models.CharField(max_length=100)
98     last_name = models.CharField(max_length=100)
99     date_of_birth = models.DateField(null=True, blank=True)
100     date_of_death = models.DateField('Died', null=True, blank=True)
101
102     class Meta:
103         ordering = ['last_name', 'first_name']
104
105     def get_absolute_url(self):
106         """Returns the URL to access a particular author instance."""
107         return reverse('author-detail', args=[str(self.id)])
108
109     def __str__(self):
110         """String for representing the Model object."""
111         return f'{self.last_name}, {self.first_name}'

```

Listing 5: Código de urls de la aplicación catalog (redireccionamientos)

```

1 from django.urls import path
2 from . import views
3

```

```
4 urlpatterns = [  
5     path('', views.index, name='index'), # name: para enlace <a href="{% url 'index'  
6     path('books/', views.BookListView.as_view(), name="books"),  
7     path('book/<int:pk>', views.BookDetailView.as_view(), name='book-detail'),  
8     path('author/', views.AuthorListView.as_view(), name='author'),  
9     path('author/<int:pk>', views.AuthorDetailView.as_view(), name='author-detail'),  
10 ]
```

Listing 6: Código de vistas (administración de contenido)

```
1  
2 from django.shortcuts import render  
3 from django.db.models import Count  
4 from django.views import generic  
5 from django.shortcuts import get_object_or_404  
6 from django.http import Http404  
7 from .models import * # Book, Author, BookInstance, Genre, Language  
8  
9 # Create your views here.  
10  
11 def index(request):  
12  
13     # Generate counts of some of the main objects  
14     num_books = Book.objects.all().count()  
15     num_instances = BookInstance.objects.all().count()  
16     num_instances_available = BookInstance.objects.filter(status__exact='a').count()  
17     num_authors = Author.objects.count() # The 'all()' is implied by default.  
18  
19     # Count genres  
20     num_genres = Genre.objects.annotate(num_books=Count('book')).count()  
21     search_word = 'Harry' # Replace 'example' with the word you want to search for  
22     num_books_with_word = Book.objects.filter(title__icontains=search_word).count()  
23  
24     # Numero de visitantes (registro de session)  
25     num_visits = request.session.get('num_visits', 0)  
26     request.session['num_visits'] = num_visits + 1  
27  
28     context = {  
29         'num_books': num_books,  
30         'num_instances': num_instances,  
31         'num_instances_available': num_instances_available,  
32         'num_authors': num_authors,  
33         'num_genres': num_genres,  
34         'num_books_with_word': num_books_with_word,  
35         'num_visits': num_visits,  
36     }  
37  
38     # Render the HTML template index.html with the data in the context variable  
39     return render(request, 'index.html', context=context)  
40  
41  
42 class BookListView(generic.ListView):  
43     model = Book # consulta de BBDD Book  
44     # context_object_name = 'book_list' # Nombre identificador para plantilla  
45     # queryset = Book.objects.filter(title__icontains='Harry')[:5]  
46     # template_name = 'catalog/book_list.html'  
47     paginate_by = 10  
48  
49     def get_queryset(self):  
50         #return Book.objects.filter(title__icontains='Harry')[:5]  
51         return Book.objects.all()  
52  
53     def get_context_data(self, **kwargs):
```

```
54     context = super(BookListView, self).get_context_data(**kwargs) # llama a BBDD
55     context['algo'] = 'agregando' # crea nuevo dato y agrega
56     return context
57
58 class BookDetailView(generic.DetailView):
59     model = Book
60
61 def book_detail_view(request, primary_key):
62     book = get_object_or_404(Book, pk=primary_key)
63     return render(request, 'catalog/book_detail.html', context={'book': book})
64
65 class AuthorListView(generic.ListView):
66     model = Author
67     paginate_by = 10
68
69 class AuthorDetailView(generic.DetailView):
70     model = Author
```

3. TAREA

- Elabore un primer informe grupal de la aplicación que desarrollará durante este semestre.
- repositorio grupal:
https://github.com/AndreRH09/Proyecto_Pweb2.git
- Utilicen todas las recomendaciones dadas en la aplicación library.

4. PREGUNTAS

- Por cada integrante del equipo, resalte un aprendizaje que adquirió al momento de estudiar Django. No se reprima de ser detallista. Coloque su nombre entre parentesis para saber que es su aporte.

En lo personal he adquirido amplio conocimiento sobre el desarrollo web con Django, abarcando aspectos como el enrutamiento, modelos y bases de datos, plantillas, formularios y seguridad. Esto les ha permitido desarrollar aplicaciones web robustas y escalables utilizando el framework Django. También Aprendí a utilizar el sistema de enrutamiento de Django para definir las URL de mi aplicación y asignar las vistas correspondientes. También comprendí la importancia de mantener una estructura de archivos organizada para facilitar el desarrollo y la localización de los recursos.

Referencias

- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website
- <https://github.com/mdn/django-locallibrary-tutorial>
- <https://github.com/rescovedoq/pw2/tree/main/labs/lab05>