

Informe de Laboratorio 05

Tema: Django Parte 01

Nota

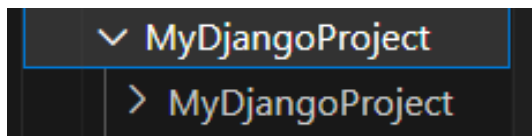
Estudiante	Escuela	Asignatura
Cristian Raul Saya Vargas, Jean Pier Valera Yana, Cristian Salhua Apfata	Escuela Profesional de Ingeniería de Sistemas	Programación Semestre: I Código: 20231001

Laboratorio	Tema	Duración
05	Django Parte 01	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	Del 10 Abril 2023	Al 17 Abril 2023

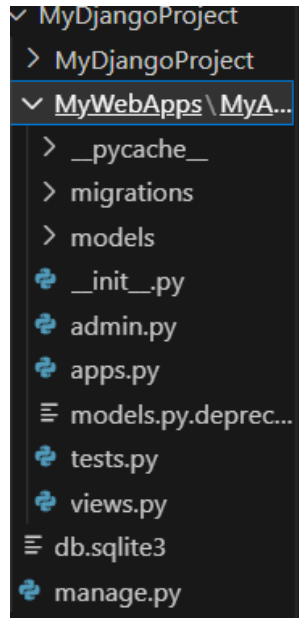
1. Tarea

- Primeramente creamos nuestro proyecto, con los comando de "django-admin startproject 'nombre del proyecto'"

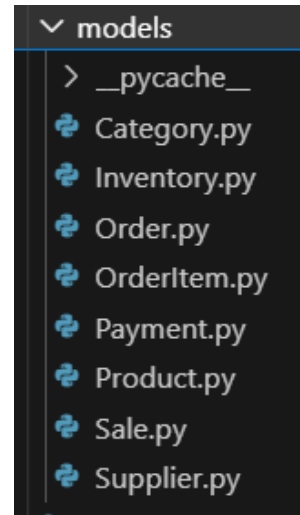


(a) creacion de proyecto

- Creandonos la siguiente rama de archivos, junto a nuestro archivo principal el manage.py
- Dentro de la subcarpeta MyDjangoProject la carpeta de models, donde se guardaran nuestros modelos para nuestra pagina



(a) Rama de archivos creados



(b) Modelos creados

- Category.py : Tenemos nuestro primer modelo el cual representa la categoria de los productos.

```
from django.db import models  Cristian Sal

"""
Modelo para representar las categorías de los
productos.
"""

class Category(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
    created_at = models.DateTimeField(
        auto_now_add=True)
    updated_at = models.DateTimeField(
        auto_now=True)

    def __str__(self):
        return self.name
```

(a) Category.py

- Order.py : Modelo que nos representa el orden de las compras
- Inventory.py : Modelo que nos representa el inventario de nuestra pagina

<pre>from django.db import models from .Product import Product """ Modelo para representar el inventario de productos. """ Cristian Salhua, last week 1 author (Cristian Salhua) class Inventory(models.Model): product = models.OneToOneField(Product, on_delete=models.CASCADE) location = models.CharField(max_length=100) last_replenishment = models.DateField() def __str__(self): return f"Inventory: {self.product}"</pre>	<pre>from django.db import models from django.contrib.auth.models import User """ Modelo para representar las órdenes de compra. """ Cristian Salhua, last week 1 author (Cristian Salhua) class Order(models.Model): user = models.ForeignKey(User, on_delete=models.CASCADE) date = models.DateTimeField(auto_now_add=True) total = models.DecimalField(max_digits=8, decimal_places=2) def __str__(self): return f"Order #{self.pk}"</pre>
--	--

(a) Inventory.py

(b) Order.py

- OrderItem.py : El siguiente modelo representa el orden de compra de los productos
- Paymet.py : El siguiente modelo representa el pago de las ventas

<pre>1 from django.db import models 2 3 from .Order import Order 4 from .Product import Product 5 6 """ 7 Modelo para representar los elementos de una 8 orden de compra. 9 """ 10 Cristian Salhua, last week 1 author (Cristian Salhua) 11 class OrderItem(models.Model): 12 order = models.ForeignKey(Order, 13 on_delete=models.CASCADE) 14 product = models.ForeignKey(Product, 15 on_delete=models.CASCADE) 16 quantity = models.PositiveIntegerField() 17 price = models.DecimalField(max_digits=8, 18 decimal_places=2) 19 20 def __str__(self): 21 return f"{self.product} - {self.quantity}"</pre>	<pre>from django.db import models from .Sale import Sale """ Modelo para representar los pagos de las ventas. """ Cristian Salhua, last week 1 author (Cristian Salhua) class Payment(models.Model): sale = models.ForeignKey(Sale, on_delete=models.CASCADE) amount = models.DecimalField(max_digits=8, decimal_places=2) payment_date = models.DateField() def __str__(self): return f"Payment #{self.pk}"</pre>
---	---

(a) OrderItem.py

(b) Paymet.py

- Product.py : El siguiente modelo representa los productos de nuestra pagina
- Sale.py : El siguiente modelo representa las ventas de nuestro producto

```
from django.db import models

from .Category import Category
from .Supplier import Supplier

"""
Modelo para representar los productos.
"""
Cristian Salhua, last week | 1 author (Cristian Salhua)
class Product(models.Model):
    name = models.CharField(
        (max_length=100)
    )
    description = models.TextField()
    price = models.DecimalField(
        (max_digits=10, decimal_places=2)
    )
    quantity = models.PositiveIntegerField(
    )
    created_at = models.DateTimeField(
        (auto_now_add=True)
    )
    updated_at = models.DateTimeField(
        (auto_now=True)
    )
    category = models.ForeignKey(
        Category, on_delete=models.CASCADE)
    supplier = models.ForeignKey(
        Supplier, on_delete=models.CASCADE)

    def __str__(self):
        return self.name
```

(a) Product.py

```
from django.db import models
from django.contrib.auth.models import User

"""
Modelo para representar las ventas.
"""
Cristian Salhua, last week | 1 author (Cristian Salhua)
class Sale(models.Model):
    customer = models.ForeignKey(User,
        on_delete=models.CASCADE)
    date = models.DateTimeField(
        (auto_now_add=True)
    )
    total = models.DecimalField(
        (max_digits=8, decimal_places=2)
    )

    def __str__(self):
        return f"Sale #{self.pk}"
```

(b) Sale.py

- Supplier.py : Tenemos nuestro ultimo modelo con los datos de proveedores

```
from django.db import models

"""
Modelo para representar los proveedores.
"""

Cristian Salhua, last week | 1 author (Cristian Salhua)
class Supplier(models.Model):
    name = models.CharField(
        max_length=100)
    address = models.CharField(
        max_length=200)
    contact_person = models.CharField(
        max_length=100)
    email = models.EmailField()
    phone_number = models.CharField(
        max_length=20)
    created_at = models.DateTimeField(
        auto_now_add=True)
    updated_at = models.DateTimeField(
        auto_now=True)

    def __str__(self):
        return self.name
```

(a) Supplier.py

2. Compilacion

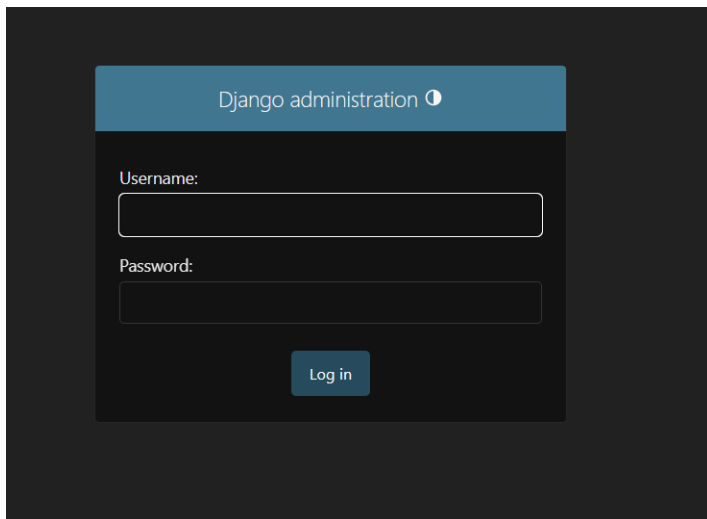
- Para el proceso de compilamiento tendremos que primero de hacer correr el sistema, para que se nos pueda crear la base de datos que en caso de python usa el SQLITE
- Creandonos asi el archibo db.sqlite
- Continuando Haremos uso del comando makemigration y el migrate para poder migrar los datos ingresados, los cuales se guardar dentro de la carpeta "migrations"
- Finalmente procederemos a compilar llamando al archivo manage.py con el comando "python manage.py runserver"

```
● Django version 4.2.2, using settings 'MyDjangoProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
●
○
```

(a) link

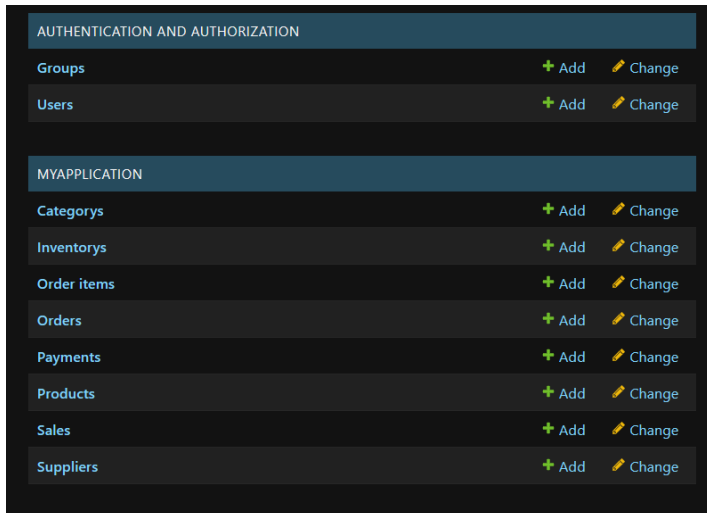
- Vemos que nos compila un link el cual se encuentra en el puerto 8000

- A continuación nos aparecerá la página para poder loguearnos, la cual nos mandará a la página donde se encuentran los modelos creados



(a) Página de logueo

- Nos aparece la página con los modelos creados en nuestro proyecto



(a) Página de logueo

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/rescobedoq/programacion.git>
- URL para el laboratorio 01 en el Repositorio GitHub.
- <https://github.com/rescobedoq/programacion/tree/main/lab01>

4. PREGUNTA DE APRENDIZAJE

- (Cristian Roberto Salhua Apfata) Al estudiar Django, he aprendido la importancia de seguir y cumplir con las convenciones y buenas prácticas de desarrollo de la plataforma. Esto incluye utilizar la estructura de directorios recomendada, aprovechar las funcionalidades y características de Django de manera adecuada, y utilizar la documentación oficial como una guía confiable durante el desarrollo de proyectos. Al seguir estas pautas, se facilita el desarrollo de aplicaciones web, así como la colaboración con otros desarrolladores.
- (Cristian Saya Vargas) El Framework Django nos facilita para realizar nuestros proyectos de una manera mas facil ya que nos brinda diferentes herramientas que optimizan el trabajo dando asi tambien la facilidad de realizar dicho proyecto en conjunto sin el riesgo de tener conflictos.
- (Jeanpier Michaelson Valera Yana) Aprendi a usar nuevas herramientas para el desarrollo con python, un nuevo framkwork que me permitia crear proyectos con una estructura mas amplica y compleja. Al utilizar Django aprendi a relacionar nuevas estructuras dentro y a manejar un siste-ma diferente entre backend y frontend. Siguiendo instrucciones observe que se crea aplicaciones o productos mucho mas profesionales.
- (Christian Henry Venero Guevara) Me ha proporcionado una base sólida para el desarrollo de aplicaciones web, brindando herramientas y estructuras que simplifican muchas tareas comunes. Es importante recordar que hay mucho más por explorar y aprender en Django, ya que es un framework en constante evolución y se pueden desarrollar aplicaciones web aún más complejas y personalizadas con él.

4.1. Estructura de laboratorio 01

- El contenido que se entrega en este laboratorio es el siguiente:

5. Pregunta: ¿Cuál es el comportamiento del algoritmo de ordenamiento por inserción?

- El algoritmo muestra un comportamiento cuadrático de $O(n^2)$.
- Se trabajaron los peores casos desde un arreglo de tamaño 1 hasta N.
- Para obtener un gráfico ideal se utilizó $N=10,000$.

6. Rúbricas

6.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

6.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4			
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4			
Total		20		12	

7. Referencias

- <https://www.w3schools.com/java/default.asp>
- <https://www.geeksforgeeks.org/insertion-sort/>