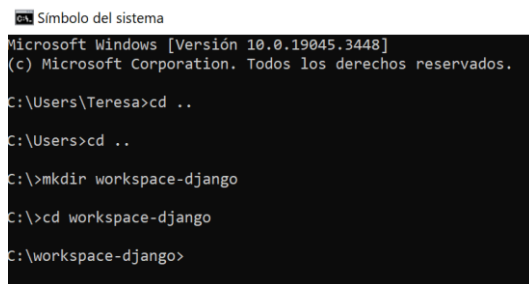


## Laboratorio – Evaluación N1 – versión 3

Crear Proyecto **Sumativa1v2** y una aplicación **App**

Pasos:

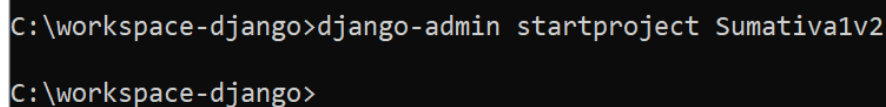
1. Abrir CMD
2. Crear un espacio de trabajo con la instrucción **mkdir workspace-django**



```
Microsoft Windows [Versión 10.0.19045.3448]
(c) Microsoft Corporation. Todos los derechos reservados.

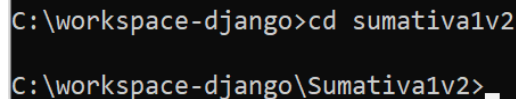
C:\Users\Teresa>cd ..
C:\Users>cd ..
C:\>mkdir workspace-django
C:\>cd workspace-django
C:\workspace-django>
```

3. Crear el proyecto **Sumativa1v2** ejecutando la siguiente instrucción:  
**django-admin startproject Sumativa1v2** o **Python -m django startproject Sumativa1v2**



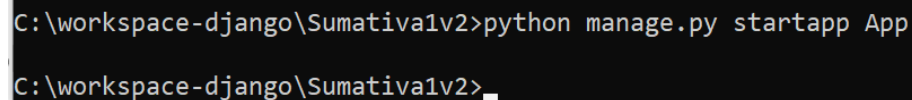
```
C:\workspace-django>django-admin startproject Sumativa1v2
C:\workspace-django>
```

4. Entrar al proyecto **Sumativa1v2**



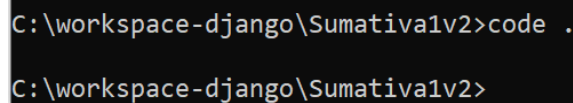
```
C:\workspace-django>cd sumativa1v2
C:\workspace-django\Sumativa1v2>
```

5. Crear la aplicación **App** ejecutando la siguiente instrucción:  
**python manage.py startapp App**



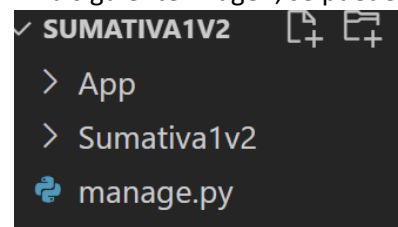
```
C:\workspace-django\Sumativa1v2>python manage.py startapp App
C:\workspace-django\Sumativa1v2>
```

6. Abrir visual code



```
C:\workspace-django\Sumativa1v2>code .
C:\workspace-django\Sumativa1v2>
```

En la siguiente imagen, se puede visualizar el proyecto **Sumativa1** y la aplicación **App**

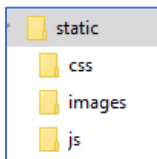


```
✓ SUMATIVA1V2
  > App
  > Sumativa1v2
  manage.py
```

7. En el proyecto crear la carpeta **templates** y **templatesApp** dentro de esta carpeta. Revisar la creación en la imagen siguiente.

```
C:\workspace-django\Sumativa1v2>mkdir templates
C:\workspace-django\Sumativa1v2>cd templates
C:\workspace-django\Sumativa1v2\templates>mkdir templatesApp
C:\workspace-django\Sumativa1v2\templates>cd..
C:\workspace-django\Sumativa1v2>_
```

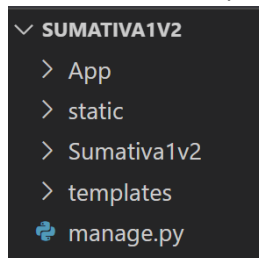
8. Para almacenar contenido estático crear la siguiente estructura de carpetas dentro del proyecto



```
C:\workspace-django\Sumativa1v2>mkdir static
C:\workspace-django\Sumativa1v2>cd static
C:\workspace-django\Sumativa1v2\static>mkdir css
C:\workspace-django\Sumativa1v2\static>mkdir js
C:\workspace-django\Sumativa1v2\static>mkdir images
C:\workspace-django\Sumativa1v2\static>_
```

9.

Nota: la carpeta static está a la misma altura de la carpeta templates. Se usan los mismos pasos de creación de carpetas del punto anterior.



10. Modificar el archivo de configuración **settings.py** del proyecto. Las modificaciones son:
- a) Incluir la aplicación **App** al Proyecto

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'App',
]
```

b) Indicar la ruta de los templates del Proyecto

Las instrucciones son:

- i. Incluir import os

```
import os
from pathlib import Path
```

- ii. Crear la variable TEMPLATE\_DIR con la ruta de la carpeta templates del Proyecto y agregarla

```
# Build paths inside the project like this: BASE_DIR
BASE_DIR = Path(__file__).resolve().parent.parent
TEMPLATE_DIR=os.path.join(BASE_DIR,'templates')
```

- iii. Especificar la variable TEMPLATE\_DIR en el código que se indica en la imagen

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.ba
        'DIRS': [TEMPLATE_DIR],
        'APP_DIRS': True,
        'OPTIONS': {
```

c) Revisar y configurar contenido estático

- i. Especificar la variable **STATICFILES\_DIRS** en el código como se indica en el recuadro amarillo. En otras palabras, agregar la variable **STATICFILES\_DIRS**

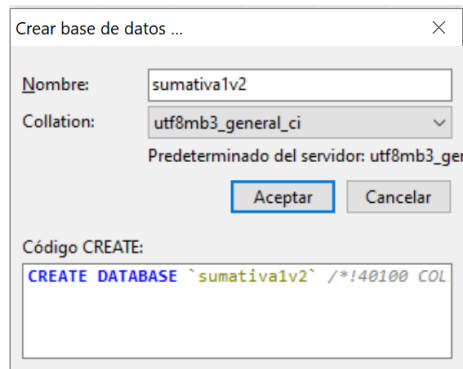
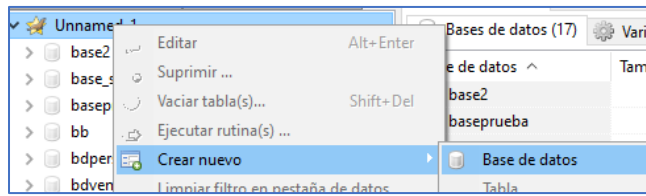
```
STATIC_URL = 'static/'
STATICFILES_DIRS=[os.path.join(BASE_DIR,'static')]
```

d) Configurar que la consola de administración sea en español

```
LANGUAGE_CODE = 'en-ES'
```

e) Crear y configurar la Base de Datos

- i. Crear la Base de Datos **servicios** en Maria DB



- ii. Instalar PyMysql si es que no está instalado

**pip install PyMySQL**

- iii. Configurar la base de datos en el archivo [settings.py](#). En otras palabras, reemplazar el siguiente código que se ilustra en la siguiente imagen

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

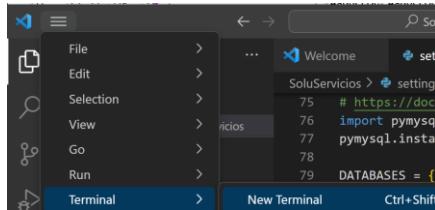
Se reemplaza por:

```
import pymysql
pymysql.install_as_MySQLdb()

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'sumativa1v2',
        'USER': 'root',
        'PASSWORD': 'root1502',
        'PORT': 3306
    }
}
```

**RECUERDA QUE EL PASSWORD EN LABORATORIO ES ADMIN O EN TU CASA ES EL QUE TU ASIGNASTE**

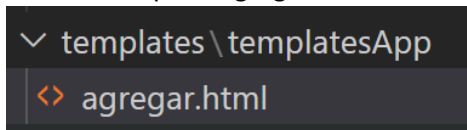
- iv. ABRE UN TERMINAL E INICIA EL SERVIDOR



```
System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly.
Migrations:
  admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 10, 2023 - 22:12:06
Django version 4.2.5, using settings 'Sumativa1.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

11. Crear el template agregar.html del Proyecto, específicamente en la carpeta `templatesApp`



12. Crear 3 clases en el archivo models.py. Las clases son: EstadReserva, TipoReserva y Reserva como se muestra a continuación.

```
App > models.py > Reserva
1  # Create your models here.
2  from django.db import models
3
4  # Create your models here.
5  class EstadReserva(models.Model):
6
7      estadoReservaId = models.CharField(primary_key = True,max_length = 3)
8      estadoReservaNombre = models.CharField(max_length = 20)
9
10     def __str__(self):
11         return "{}".format(self.estadoReservaNombre)
12
13     class TipoReserva(models.Model):
```

```

12
13 class TipoReserva(models.Model):
14
15     tipoSolicitudId = models.CharField(primary_key = True,max_length = 3)
16     tipoSolicitud = models.CharField(max_length = 20)
17
18     def __str__(self):
19         return "{}".format(self.tipoSolicitud)
20
21
22 class Reserva(models.Model):

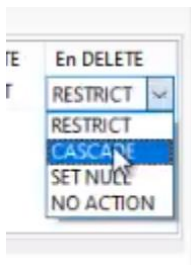
```

```

22 class Reserva(models.Model):
23
24     idSolicitud = models.AutoField(primary_key=True)
25     nombre = models.CharField(max_length=50)
26     telefono = models.CharField(max_length=12)
27     fechareserva = models.DateField()
28     horareserva = models.TimeField()
29     cantidadpersonas = models.IntegerField()
30
31     observaciones = models.CharField(max_length=5000)
32
33     website = models.URLField()
34     email = models.EmailField()
35     donante = models.BooleanField()
36     edad = models.IntegerField()
37
38     estadoReservaId = models.ForeignKey(EstadReserva, null = True, blank = False, on_delete = models.RESTRICT)
39     tipoSolicitudId = models.ForeignKey(TipoReserva, null = True, blank = False, on_delete = models.RESTRICT)
40
41

```

Revise o estudie la propiedad on\_delete. En mariadb se tienen estas opciones.



13. Crear archivo forms.py como se indica a continuación.

```
App > forms.py > ReservaForm > clean_edad
1  from django import forms
2  from .models import Reserva, EstadReserva, TipoReserva
3
4  class ReservaForm(forms.ModelForm):
5      class Meta:
6          model = Reserva
7          fields = '__all__'
8
9      ESTADOS_CHOICES = (
10         ('guardado', 'GUARDADO'),
11         ('anulado', 'ANULADO'),
12         ('confirmado', 'CONFIRMADO'),
13     )
14
15     fechareserva = forms.DateField(widget=forms.widgets.DateInput(attrs={'type': 'date'}), label='Fecha Reserva')
16     horasreserva = forms.TimeField(widget=forms.widgets.TimeInput(attrs={'type': 'time', 'format': '%H:%M'}), label='Hora Reserva')
17     observaciones=forms.CharField(widget=forms.Textarea(attrs={'rows': 4, 'cols': 50}))
18     cantidadpersonas=forms.IntegerField(label='Cantidad de Personas')
19     #Estado reserva
20     estadoReservaId = forms.ModelChoiceField(queryset=EstadReserva.objects.all(), label='Estado Reserva')
21     estadoReservaId.widget.attrs['class'] = 'form-select'
22
23     #Tipo reserva
24     tipoSolicitudId = forms.ModelChoiceField(queryset=TipoReserva.objects.all(), label='Tipo Reserva')
25     tipoSolicitudId.widget.attrs['class'] = 'form-select'
26
27     def clean_edad(self):
28         edad = self.cleaned_data.get('edad')
```

```
App > forms.py > ...
24     tipoSolicitudId = forms.ModelChoiceField(queryset=TipoReserva.objects.all(), label='Tipo Reserva')
25     tipoSolicitudId.widget.attrs['class'] = 'form-select'
26
27     def clean_edad(self):
28         edad = self.cleaned_data.get('edad')
29         if edad < 18:
30             raise forms.ValidationError("Debe ser mayor de edad para reservar")
31         return edad
32
33     def clean_nombre(self):
34         nombre = self.cleaned_data.get('nombre')
35         if len(nombre) <= 2:
36             raise forms.ValidationError("El nombre debe tener más de dos letras.")
37         return nombre
38
39     def clean_observaciones(self):
40         observaciones = self.cleaned_data.get('observaciones')
41         palabras = observaciones.split()
42
43         if len(palabras) < 5:
44             raise forms.ValidationError("Las observaciones deben contener al menos 5 palabras.")
45
46         return observaciones
47
```

14. Copiar el template **agregar.html** del proyecto anterior sumativa1

```
templates > templatesApp > agregar.html
1  <!DOCTYPE html>
2  {% load static %}
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha
9      <title>ADM. RESERVAS</title>
10     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">
11 </head>
12 <body class="container mt-5">
13     <div class="alert alert-info display-4 text-center">ADMINISTRACIÓN DE RESERVAS</div>
14     <!--Formulario-->
15     <form method="POST">
16         <table class="table">
17             {{ form }}
18             {% csrf_token %}
19             <tr>
20                 <td colspan="2">
21                     <input type="submit" class="btn btn-success float-end" value="Guardar">
22                 </td>
23             </tr>
24         </table>
25     </form>
26
27     {% if reservas %}
28     <table class="table">
29         <thead>
30             <tr>
31                 <th>Nombre</th>
32                 <th>Telefono</th>
33                 <th>Fecha Reserva</th>
34                 <th>Hora Reserva</th>
35                 <th>Website</th>
36                 <th>Estado</th>
37                 <th>Email</th>
38                 <th>---</th>
39             </tr>
40         </thead>
```



```

templates > templatesApp > agregar.html
38     <th>---</th>
39   </tr>
40 </thead>
41 <tbody>
42   {% for reserva in reservas %}
43   <tr>
44     <td>{{ reserva.nombre }}</td>
45     <td>{{ reserva.telefono }}</td>
46     <td>{{ reserva.fechareserva }}</td>
47     <td>{{ reserva.horareserva }}</td>
48     <td>{{ reserva.website }}</td>
49     <td>{{ reserva.estadoReservaId }}</td>
50     <td>{{ reserva.email }}</td>
51     <td>
52       <a class="btn btn-success btn-small" href="../../actualizarReserva/{{reserva.idSolicitud}}">
53         <i class="bi bi-pencil-fill"></i>
54       </a>
55       <a class="btn btn-danger btn-small" href="../../eliminarReserva/{{reserva.idSolicitud}}">
56         <i class="bi bi-trash-fill"></i>
57       </a>
58     </td>
59   </tr>
60   {% endfor %}
61 </tbody>
62 </table>
63 {% else %}
64 <div class="alert alert-info">No se encuentran reservas en el Sistema</div>
65 {% endif %}
66
67 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-Hwwwvtg8No3bZJJLYd8oVXjrI"
68 </body>
69 </html>

```

15. Copiar el archivo **views.py** del proyecto anterior.

```

App > views.py > ...
1  from django.shortcuts import render, redirect
2
3  # Create your views here.
4  from .models import Reserva
5  from .forms import ReservaForm
6

```

```

def agregarReserva(request):
    form = ReservaForm()
    if request.method == 'POST':
        form = ReservaForm(request.POST)
        if form.is_valid():
            form.save()
            form = ReservaForm()

    reservas = Reserva.objects.all()
    data = {'form': form, 'reservas': reservas}
    return render(request, 'templatesApp/agregar.html', data)

```

Se modifica con respecto al proyecto anterior. Se cambia el id por idSolicitud (línea 21)

```
19
20 def eliminarReserva(request, id):
21     reserva = Reserva.objects.get(idSolicitud=id)
22     reserva.delete()
23     return redirect('/')
24
```

Se modifica con respecto al proyecto anterior. Se cambia el id por idSolicitud (línea 26)

```
25 def actualizarReserva(request, id):
26     reserva = Reserva.objects.get(idSolicitud=id)
27     form = ReservaForm(instance=reserva)
28     if (request.method == 'POST'):
29         form = ReservaForm(request.POST, instance=reserva)
30         if form.is_valid():
31             form.save()
32             form = ReservaForm()
33
34     reservas = Reserva.objects.all()
35     data = {'form': form, 'reservas': reservas}
36     return render(request, 'templatesApp/agregar.html', data)
```

16. Copiar el archivo [urls.py](#) del proyecto anterior

```
from django.contrib import admin
from django.urls import path
from App import views

urlpatterns = [
    path('', views.agregarReserva),
    path('eliminarReserva/<int:id>', views.eliminarReserva),
    path('actualizarReserva/<int:id>', views.actualizarReserva),
    path('admin/', admin.site.urls),
]
```

17. Realizar makemigrations

```
PS C:\workspace-django\Sumativa1v2> python manage.py makemigrations
Migrations for 'App':
  App\migrations\0001_initial.py
    - Create model EstadReserva
    - Create model TipoReserva
    - Create model Reserva
PS C:\workspace-django\Sumativa1v2> |
```

## 18. Realizar migrate

```
PS C:\workspace-django\Sumativa1v2> python manage.py migrate
Operations to perform:
  Apply all migrations: App, admin, auth, contenttypes, sessions
Running migrations:
  Applying App.0001_initial... OK
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
PS C:\workspace-django\Sumativa1v2>
```

## 19. Modificar la base de datos. Ingresar datos en la tabla estadoreserva y tiporeserva.

sumativa1v2.app\_estadreserva: 3 filas en total (aproximadamente)

estadoReservald	estadoReservaNombre
1	pendiente
2	confirmada
3	anulada

sumativa1v2.app\_tiporeserva: 2 filas en total (aproximadamente)





tipoSolicitudId	tipoSolicitud
1	consulta dental
2	consulta medica

## 20. Iniciar el servidor y revisar

```
PS C:\workspace-django\Sumativa1v2> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 12, 2023 - 11:22:19
Django version 4.2.5, using settings 'Sumativa1v2.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

## ADMINISTRACIÓN DE RESERVAS





Nombre:	<input type="text"/>
Telefono:	<input type="text"/>
Fecha Reserva:	<input type="text" value="dd-mm-aaaa"/> 
Hora Reserva:	<input type="text" value="--:--"/> 
Cantidad de Personas:	<input type="text"/>
Observaciones:	<div><div></div></div>
Website:	<input type="text"/>
Email:	<input type="text"/>
Donante:	<input type="checkbox"/>
Edad:	<input type="text"/>
Estado Reserva:	<input type="text" value="-----"/> 
Tipo Reserva:	<input type="text" value="-----"/> 



Guardar

No se encuentran reservas en el Sistema

21. Pruebe agregando un registro y eliminándolo.

## ADMINISTRACIÓN DE RESERVAS

Nombre:	<input type="text"/>
Telefono:	<input type="text"/>
Fecha Reserva:	<input type="text" value="dd-mm-aaaa"/> 
Hora Reserva:	<input type="text" value="--:--"/> 
Cantidad de Personas:	<input type="text"/>
Observaciones:	<div><div></div></div>
Website:	<input type="text"/>
Email:	<input type="text"/>
Donante:	<input type="checkbox"/>
Edad:	<input type="text"/>
Estado Reserva:	<input type="text" value="-----"/> 
Tipo Reserva:	<input type="text" value="-----"/> 
<div>Guardar</div>	

Nombre	Telefono	Fecha Reserva	Hora Reserva	Website	Estado	Email	---
tere	994503051	Oct. 11, 2023	1:58 p.m.	http://www.al.cl	confirmada	teresa.tapia02@inacapmail.cl	 
tere	994503051	Oct. 11, 2023	1:58 p.m.	http://www.al.cl	pendiente	teresa.tapia02@inacapmail.cl	