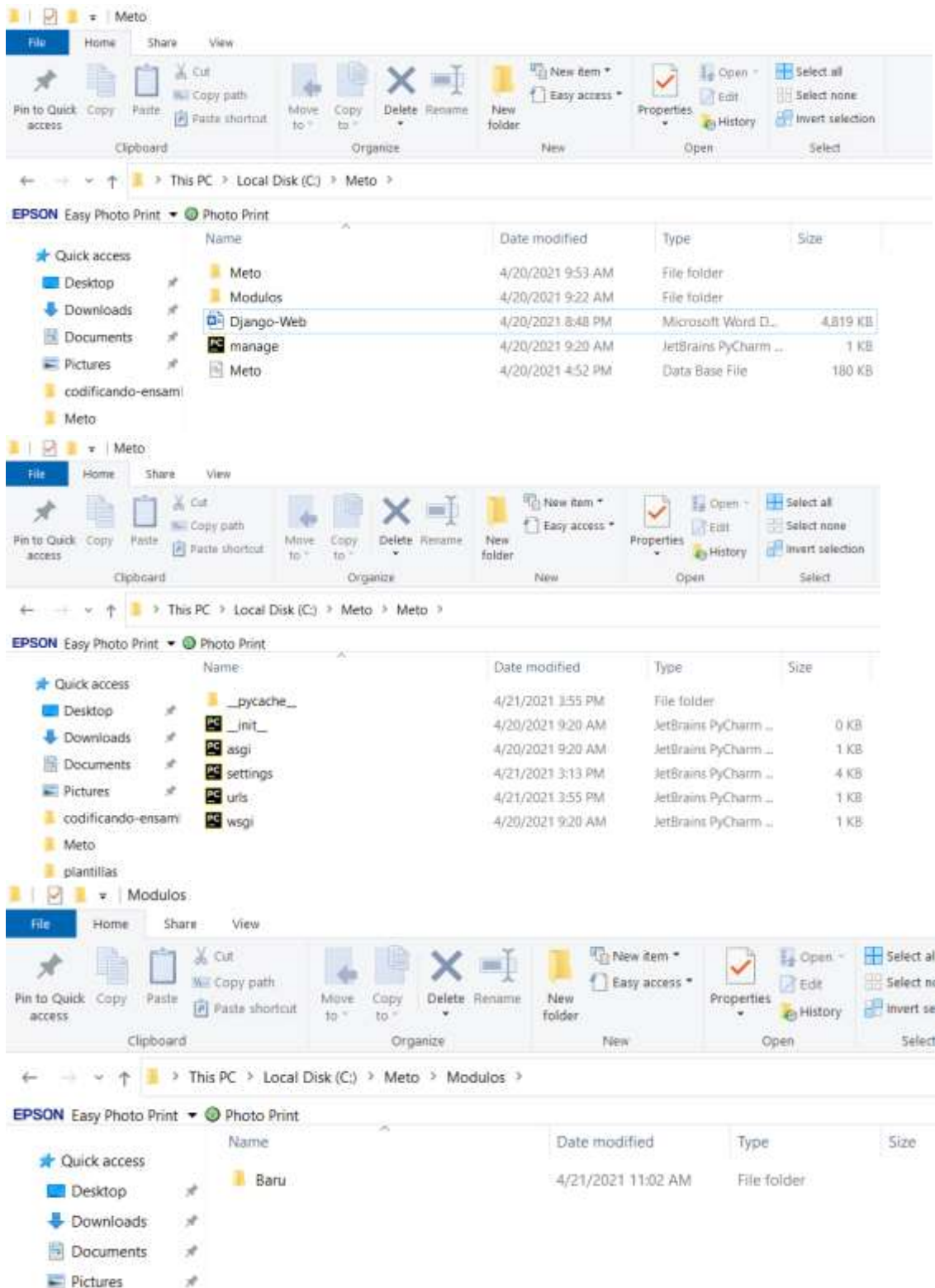
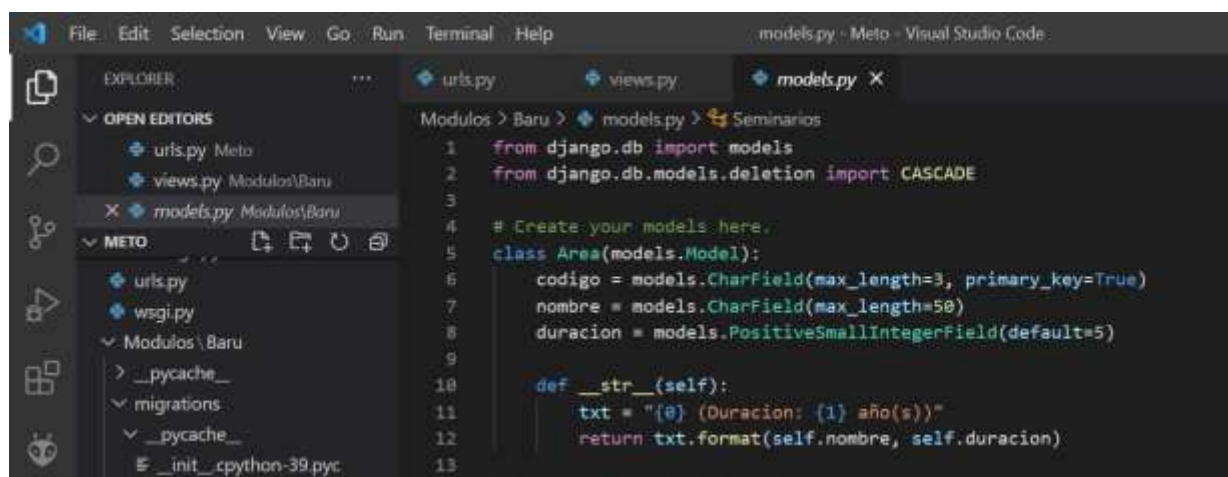
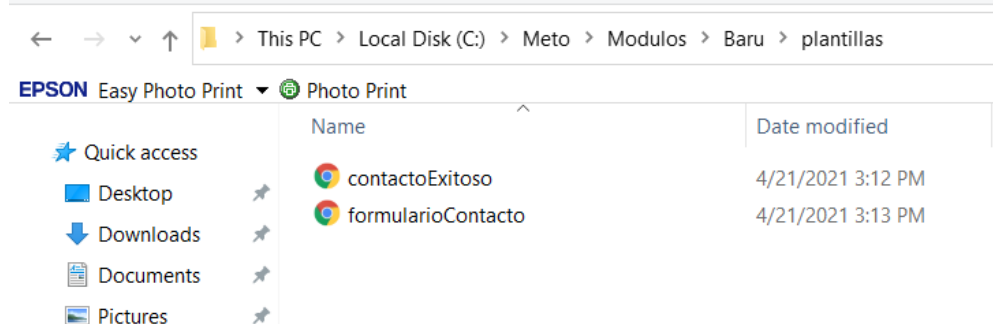
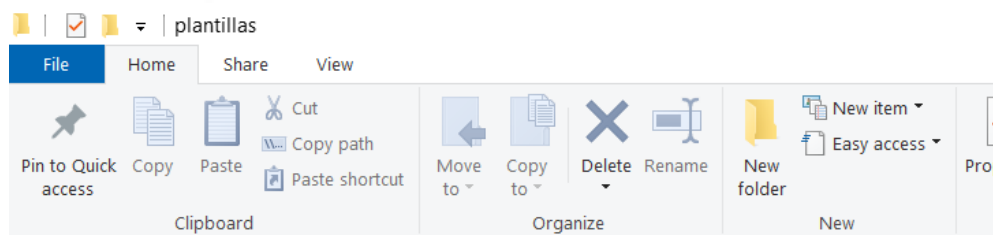
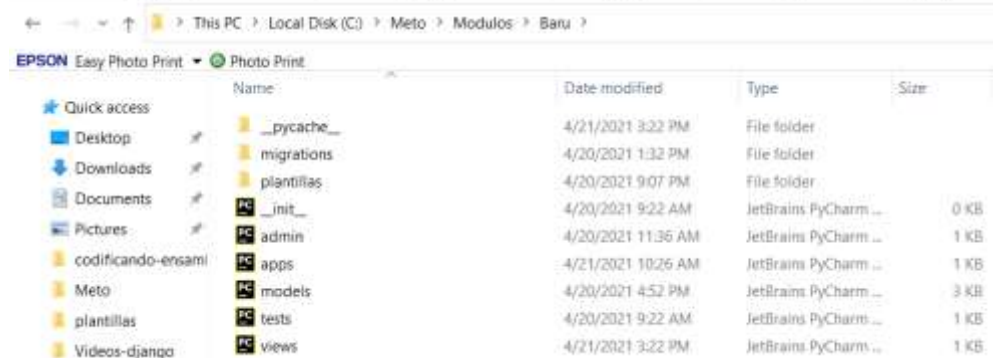
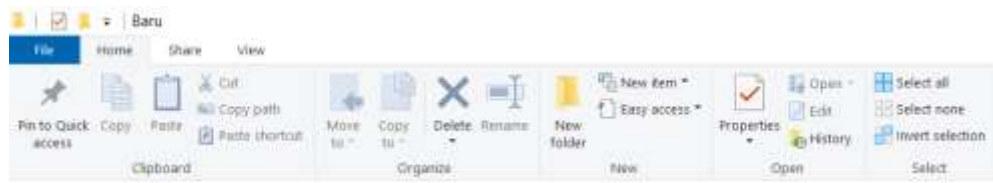


## Lab-Trabajando con Django

En esta ocasión se enviará un correo por medio de Django





```
File Edit Selection View Go Run Terminal Help models.py - Metro - Visual Studio Code

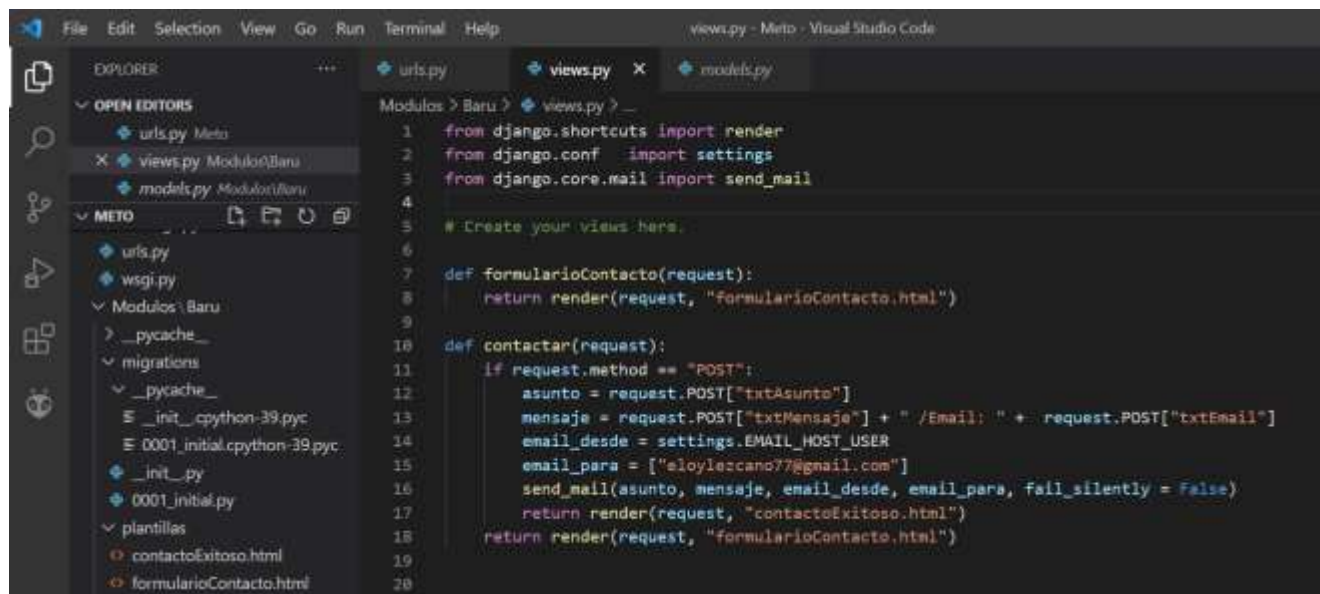
EXPLORER
OPEN EDITORS
  urls.py Metro
  views.py Modulos\Baru
  X models.py Modulos\Ularu
METO
  urls.py
  wsgi.py
  Modulos\Baru
    > __pycache__
    migrations
    > __pycache__
      E __init__.cpython-39.pyc
      E 0001_initial.cpython-39.pyc
    > __init__.py
    > 0001_initial.py
  plantillas
    contactoExitoso.html
    formularioContacto.html
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py

14 class Estudiante(models.Model):
15     dni = models.CharField(max_length=12, primary_key=True)
16     apellidoPaterno = models.CharField(max_length=15)
17     apellidoMaterno = models.CharField(max_length=15)
18     nombres = models.CharField(max_length=20)
19     fechaNacimiento = models.DateField()
20     sexos = [
21         ('F', 'Femenino'),
22         ('M', 'Masculino')
23     ]
24     sexo = models.CharField(max_length=1, choices=sexos, default='F')
25     area = models.ForeignKey(Area, null=False, blank=False, on_delete=models.CASCADE)
26     vigencia = models.BooleanField(default=True)
27
28     def nombreCompleto(self):
29         txt = "{0}, {1}, {2}"
30         return txt.format(self.apellidoPaterno, self.apellidoMaterno, self.nombres)
31
32     def __str__(self):
33         txt = "{0} / area: {1} / {2}"
34         if self.vigencia:
35             estadoEstudiante = "VIGENTE"
36         else:
37             estadoEstudiante = "DE BAJA"
38         return txt.format(self.nombreCompleto(), self.area, estadoEstudiante)
```

```
File Edit Selection View Go Run Terminal Help models.py - Metro - Visual Studio Code

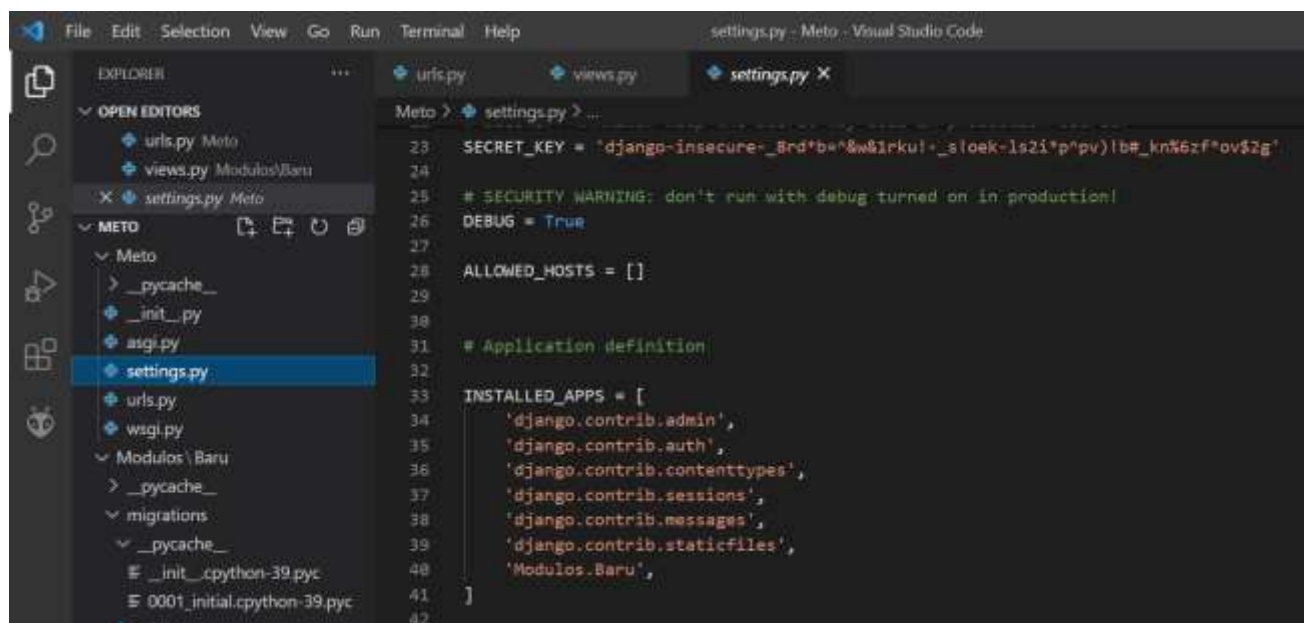
EXPLORER
OPEN EDITORS
  urls.py Metro
  views.py Modulos\Baru
  X models.py Modulos\Ularu
METO
  urls.py
  wsgi.py
  Modulos\Baru
    > __pycache__
    migrations
    > __pycache__
      E __init__.cpython-39.pyc
      E 0001_initial.cpython-39.pyc
    > __init__.py
    > 0001_initial.py
  plantillas
    contactoExitoso.html
    formularioContacto.html
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
    - $b-Correo-Django.docx

39
40 class Seminarios(models.Model):
41     codigo = models.CharField(max_length=6, primary_key=True)
42     nombre = models.CharField(max_length=30)
43     credits = models.PositiveIntegerField()
44     instructor = models.CharField(max_length=30)
45
46     def __str__(self):
47         txt = "{0} ({1}) / instructor: {2}"
48         return txt.format(self.nombre, self.codigo, self.instructor)
49
50 class Matricula(models.Model):
51     id = models.AutoField(primary_key=True)
52     estudiante = models.ForeignKey(Estudiante, null=False, blank=False, on_delete=CASCADE)
53     seminario = models.ForeignKey(Seminarios, null=False, blank=False, on_delete=CASCADE)
54     fechaMatricula = models.DateTimeField(auto_now_add=True)
55
56     def __str__(self):
57         txt = "{0} matriculado(1) en el seminario {2} / fecha: {3}"
58         if self.estudiante.sexo == ('F'):
59             letraSexo = "a"
60         else:
61             letraSexo = "o"
62         FecMat = self.fechaMatricula.strftime("%A %d/%m/%Y %H:%M:%S")
63         return txt.format(self.estudiante.nombreCompleto(), letraSexo, self.seminario, FecMat)
64
65
66
```



The screenshot shows the Visual Studio Code interface with the 'views.py' file open. The Explorer sidebar on the left shows the project structure, including 'urls.py', 'wsgi.py', and a 'Modulos' directory containing 'Baru'. The 'views.py' file contains the following code:

```
1 from django.shortcuts import render
2 from django.conf import settings
3 from django.core.mail import send_mail
4
5 # Create your views here.
6
7 def formularioContacto(request):
8     return render(request, "formularioContacto.html")
9
10 def contactar(request):
11     if request.method == "POST":
12         asunto = request.POST["txtAsunto"]
13         mensaje = request.POST["txtMensaje"] + " /Email: " + request.POST["txtEmail"]
14         email_desde = settings.EMAIL_HOST_USER
15         email_para = ["eloylezcano77@gmail.com"]
16         send_mail(asunto, mensaje, email_desde, email_para, fail_silently = False)
17         return render(request, "contactoExitoso.html")
18     return render(request, "formularioContacto.html")
19
20
```



The screenshot shows the Visual Studio Code interface with the 'settings.py' file open. The Explorer sidebar on the left shows the project structure, including 'urls.py', 'wsgi.py', and a 'Modulos' directory containing 'Baru'. The 'settings.py' file contains the following code:

```
23 SECRET_KEY = 'django-insecure-8rd*b~^&w&lrku!+_sloek-ls2i*p*pv)lb#_kn%6zf*ov$2g'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30 # Application definition
31
32 INSTALLED_APPS = [
33     'django.contrib.admin',
34     'django.contrib.auth',
35     'django.contrib.contenttypes',
36     'django.contrib.sessions',
37     'django.contrib.messages',
38     'django.contrib.staticfiles',
39     'Modulos.Baru',
40 ]
41
42
```



```
File Edit Selection View Go Run Terminal Help settings.py - Meto - Visual Studio Code

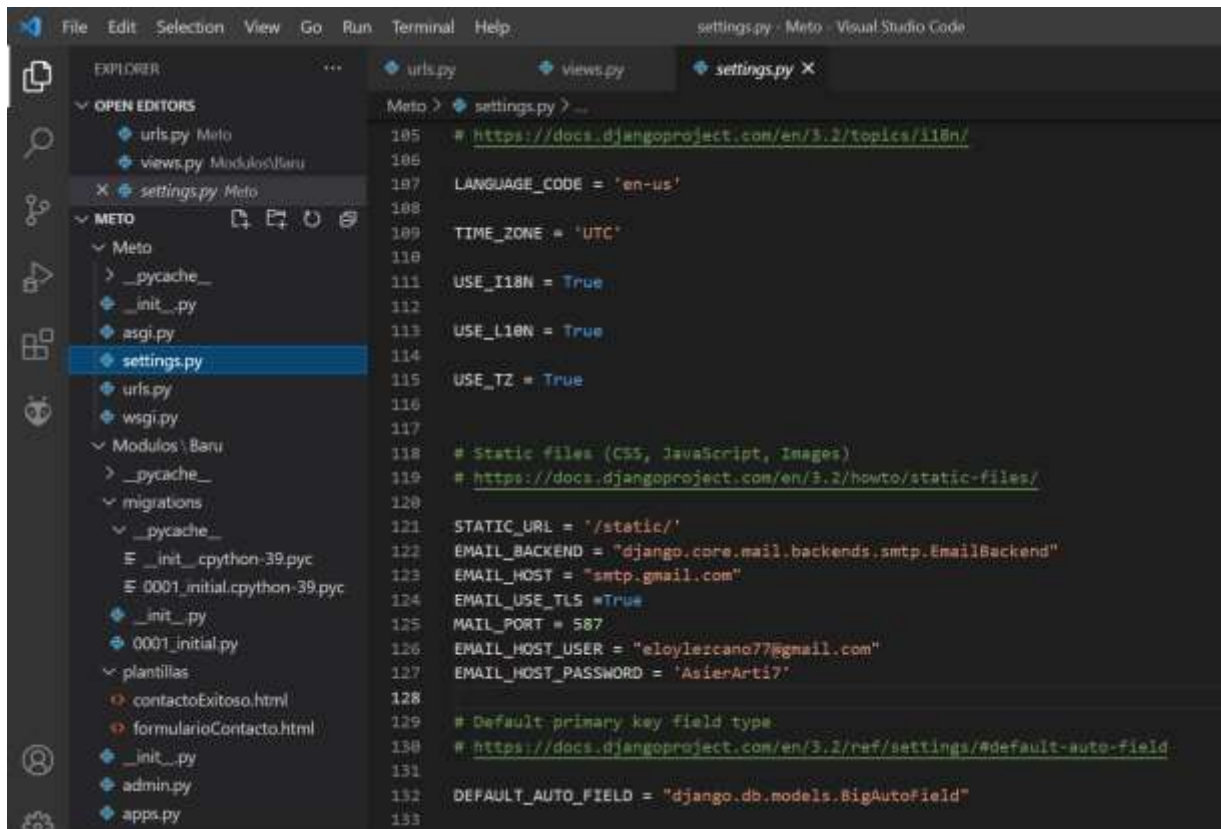
EXPLORER
OPEN EDITORS
  urls.py Meto
  views.py Modulos\Baru
  settings.py Meto
METO
  Meto
    _pycache_
    _init_.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  Modulos\Baru
    _pycache_
    migrations
      _pycache_
      _init_.cpython-39.pyc
      0001_initial.cpython-39.pyc
    _init_.py
    0001_initial.py
    plantillas
      contactoExitoso.html
      formularioContacto.html
    _init_.py

Meto > settings.py > ...
46 'django.middleware.common.CommonMiddleware',
47 'django.middleware.csrf.CsrfViewMiddleware',
48 'django.contrib.auth.middleware.AuthenticationMiddleware',
49 'django.contrib.messages.middleware.MessageMiddleware',
50 'django.middleware.clickjacking.XFrameOptionsMiddleware',
51 ]
52
53 ROOT_URLCONF = 'Meto.urls'
54
55 TEMPLATES = [
56     {
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',
58         'DIRS': ['C:/Meto/Modulos/Baru/plantillas'],
59         'APP_DIRS': True,
60         'OPTIONS': {
61             'context_processors': [
62                 'django.template.context_processors.debug',
63                 'django.template.context_processors.request',
64                 'django.contrib.auth.context_processors.auth',
65                 'django.contrib.messages.context_processors.messages',
66             ],
67         },
68     },
69 ]
70
71 WSGI_APPLICATION = 'Meto.wsgi.application'
```

```
File Edit Selection View Go Run Terminal Help settings.py - Meto - Visual Studio Code

EXPLORER
OPEN EDITORS
  urls.py Meto
  views.py Modulos\Baru
  settings.py Meto
METO
  Meto
    _pycache_
    _init_.py
    asgi.py
    settings.py
    urls.py

Meto > settings.py > ...
74
75 # Database
76 # https://docs.djangoproject.com/en/3.2/ref/settings/#databases
77
78 DATABASES = {
79     'default': {
80         'ENGINE': 'django.db.backends.sqlite3',
81         'NAME': 'Meto.db',
82     },
83 }
```

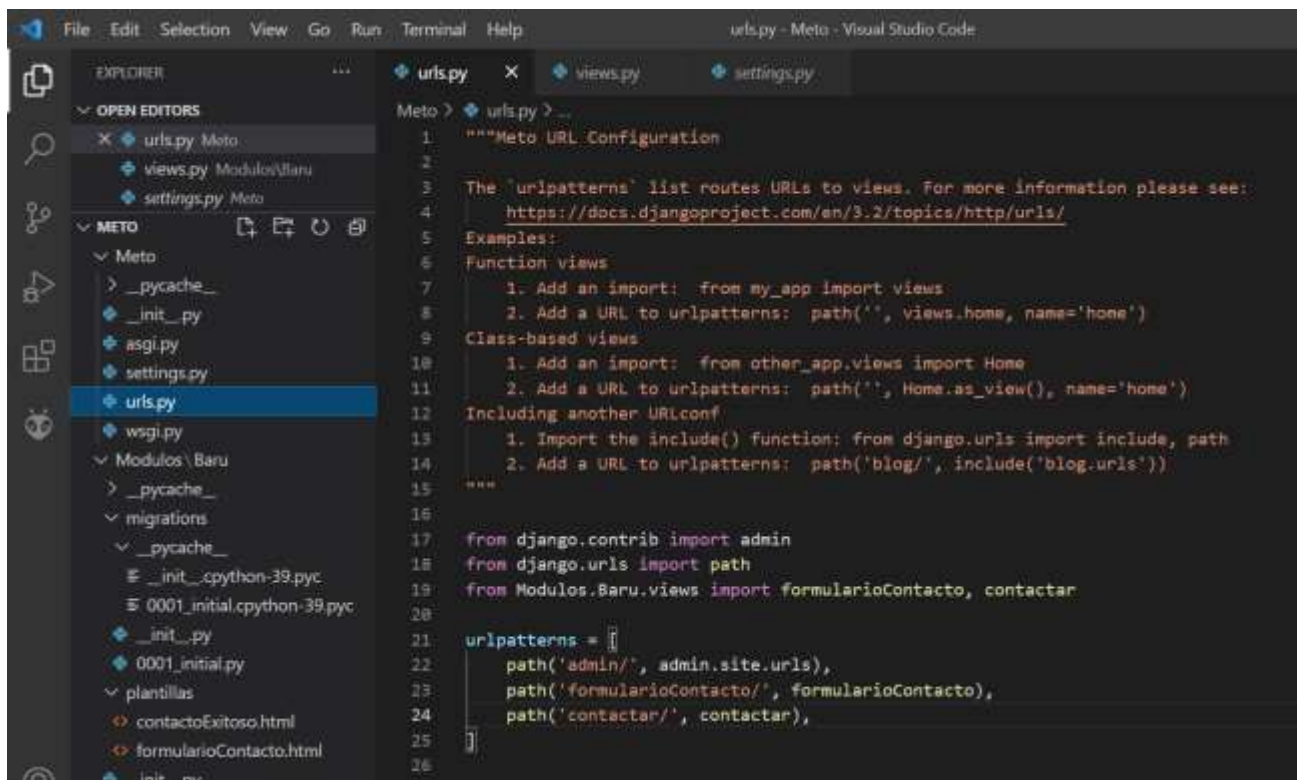


```
File Edit Selection View Go Run Terminal Help settings.py - Meto - Visual Studio Code

EXPLORER
OPEN EDITORS
  urls.py Meto
  views.py Modulos\Baru
  settings.py Meto
METO
  Meto
    _pycache_
    _init_.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  Modulos\Baru
    _pycache_
    migrations
      _pycache_
      _init_.cpython-39.pyc
      0001_initial.cpython-39.pyc
      _init_.py
      0001_initial.py
    plantillas
      contactoExitoso.html
      formularioContacto.html
      _init_.py
      admin.py
      apps.py

urls.py
views.py
settings.py X

Meto > settings.py > ...
185 # https://docs.djangoproject.com/en/3.2/topics/i18n/
186
187 LANGUAGE_CODE = 'en-us'
188
189 TIME_ZONE = 'UTC'
190
191 USE_I18N = True
192
193 USE_L10N = True
194
195 USE_TZ = True
196
197 # Static files (CSS, JavaScript, Images)
198 # https://docs.djangoproject.com/en/3.2/howto/static-files/
199
200 STATIC_URL = '/static/'
201 EMAIL_BACKEND = "django.core.mail.backends.smtp.EmailBackend"
202 EMAIL_HOST = "smtp.gmail.com"
203 EMAIL_USE_TLS = True
204 MAIL_PORT = 587
205 EMAIL_HOST_USER = "eloylercano77@gmail.com"
206 EMAIL_HOST_PASSWORD = 'AsierArti7'
207
208 # Default primary key field type
209 # https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field
210
211 DEFAULT_AUTO_FIELD = "django.db.models.BigAutoField"
```

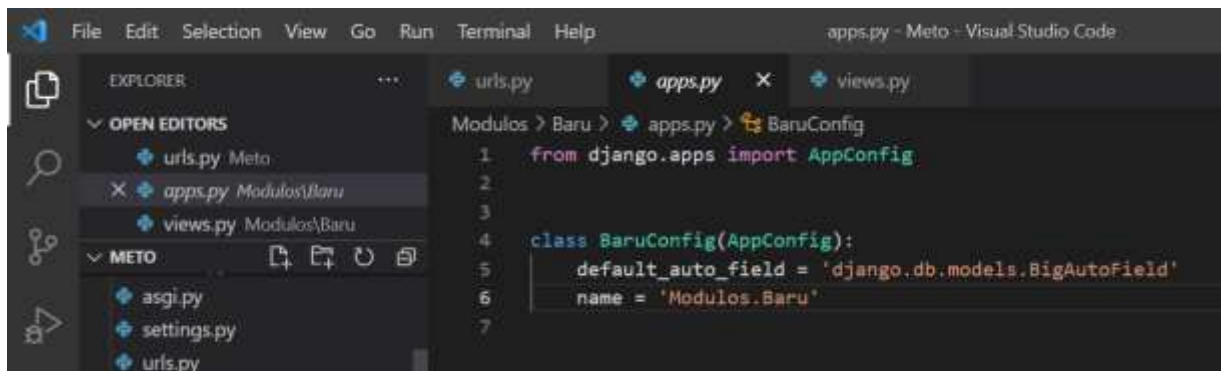


```
File Edit Selection View Go Run Terminal Help urls.py - Meto - Visual Studio Code

EXPLORER
OPEN EDITORS
  urls.py Meto
  views.py Modulos\Baru
  settings.py Meto
METO
  Meto
    _pycache_
    _init_.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  Modulos\Baru
    _pycache_
    migrations
      _pycache_
      _init_.cpython-39.pyc
      0001_initial.cpython-39.pyc
      _init_.py
      0001_initial.py
    plantillas
      contactoExitoso.html
      formularioContacto.html
      _init_.py
      admin.py
      apps.py

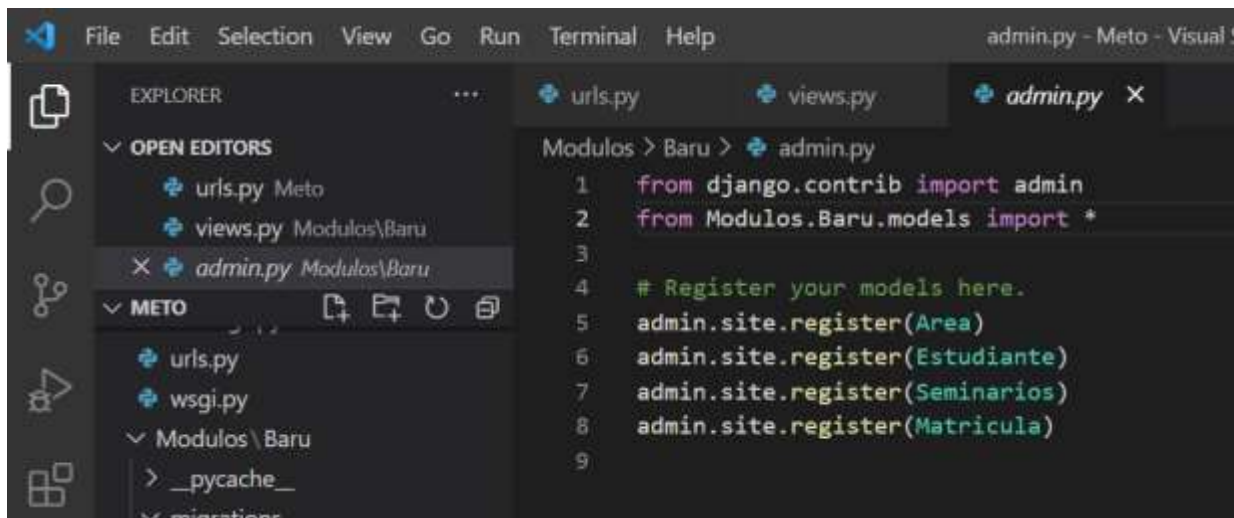
urls.py X
views.py
settings.py

Meto > urls.py > ...
1 """Meto URL Configuration
2
3 The 'urlpatterns' list routes URLs to views. For more information please see:
4     https://docs.djangoproject.com/en/3.2/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16
17 from django.contrib import admin
18 from django.urls import path
19 from Modulos.Baru.views import formularioContacto, contactar
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('formularioContacto/', formularioContacto),
24     path('contactar/', contactar),
25 ]
26
```



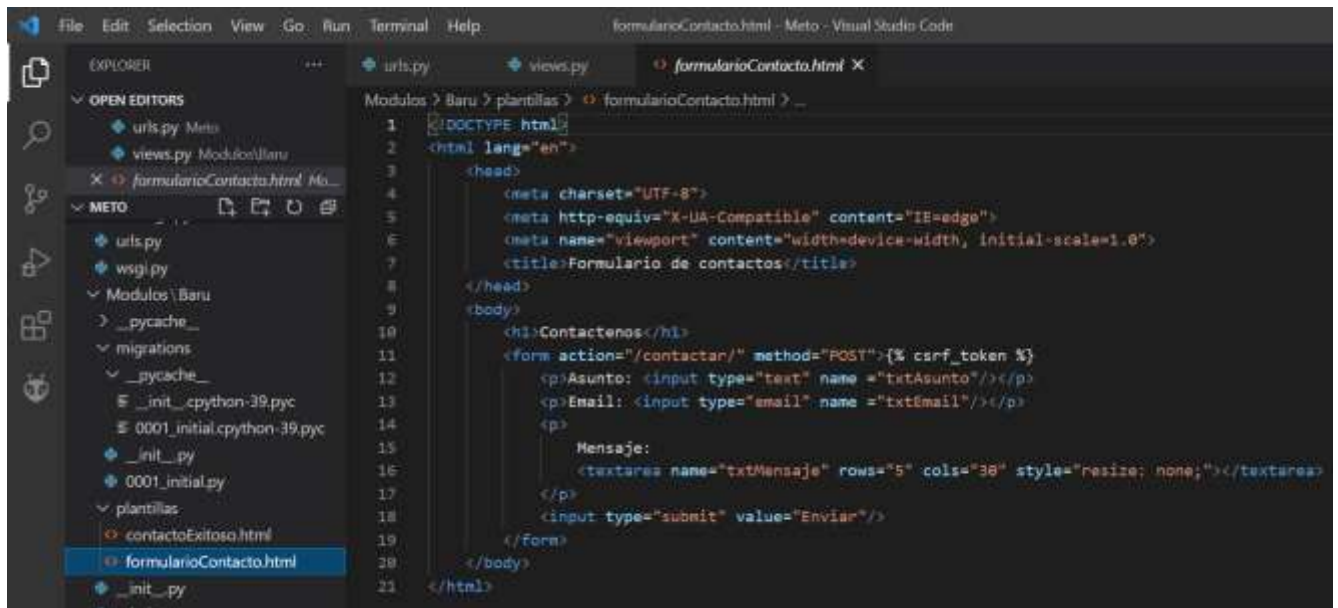
This screenshot shows the Visual Studio Code interface with the Django Baru app configuration. The Explorer sidebar on the left shows the project structure with 'apps.py' selected under 'Modulos\Baru'. The main editor displays the 'apps.py' file, which imports 'AppConfig' from 'django.apps' and defines the 'BaruConfig' class. The class inherits from 'AppConfig' and sets the 'default\_auto\_field' to 'django.db.models.BigAutoField' and the 'name' to 'Modulos.Baru'.

```
1 from django.apps import AppConfig
2
3
4 class BaruConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'Modulos.Baru'
7
```



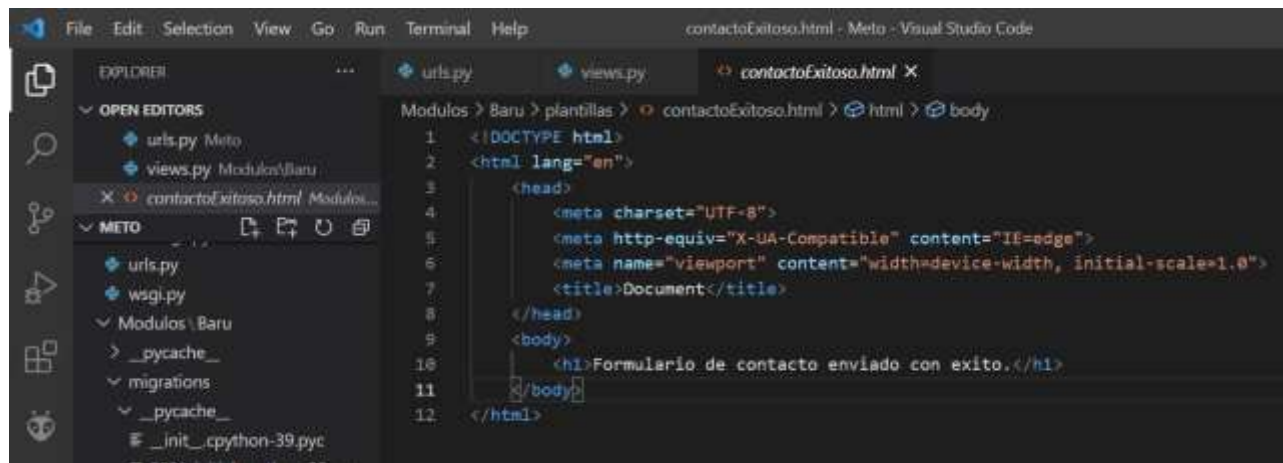
This screenshot shows the Visual Studio Code interface with the Django Baru admin configuration. The Explorer sidebar on the left shows the project structure with 'admin.py' selected under 'Modulos\Baru'. The main editor displays the 'admin.py' file, which imports 'admin' from 'django.contrib' and registers the 'Area', 'Estudiante', 'Seminarios', and 'Matricula' models with the Django admin site.

```
1 from django.contrib import admin
2 from Modulos.Baru.models import *
3
4 # Register your models here.
5 admin.site.register(Area)
6 admin.site.register(Estudiante)
7 admin.site.register(Seminarios)
8 admin.site.register(Matricula)
9
```



This screenshot shows the Visual Studio Code interface with the Django Baru contact form template. The Explorer sidebar on the left shows the project structure with 'formularioContacto.html' selected under 'plantillas'. The main editor displays the 'formularioContacto.html' file, which is an HTML template for a contact form. It includes a DOCTYPE declaration, a head section with meta tags for charset, http-equiv, and viewport, and a body section with a form for contacting. The form has a title 'Formulario de contactos' and a submit button 'Enviar'.

```
1 <!DOCTYPE html>
2 <html lang="es">
3     <head>
4         <meta charset="UTF-8">
5         <meta http-equiv="X-UA-Compatible" content="IE=edge">
6         <meta name="viewport" content="width=device-width, initial-scale=1.0">
7         <title>Formulario de contactos</title>
8     </head>
9     <body>
10         <h1>Contactenos</h1>
11         <form action="/contactar/" method="POST">{% csrf_token %}
12             <p>Asunto: <input type="text" name="txtAsunto"/></p>
13             <p>Email: <input type="email" name="txtEmail"/></p>
14             <p>
15                 Mensaje:
16                 <textarea name="txtMensaje" rows="5" cols="38" style="resize: none;"></textarea>
17             </p>
18             <input type="submit" value="Enviar"/>
19         </form>
20     </body>
21 </html>
```



The screenshot shows the Visual Studio Code editor with the file `contactoExitoso.html` open. The Explorer sidebar on the left shows the project structure with files like `urls.py`, `views.py`, and `contactoExitoso.html`. The main editor area displays the HTML code for the success page:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
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     <title>Document</title>
9   </head>
10  <body>
11    <h1>Formulario de contacto enviado con exito.</h1>
12  </body>
13 </html>
```



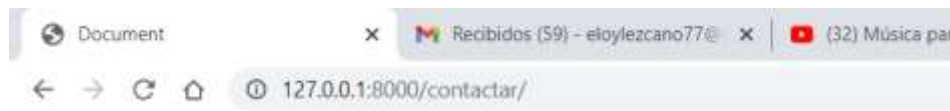
## Contactenos

Asunto: Base de Datos en Django

Email: eloylezcano77@gmail.com

Mensaje: Aquí todo bien.

Enviar



**Formulario de contacto enviado con exito.**



