



Basado en Curso Django de Píldoras Informáticas:

<https://www.youtube.com/watch?v=7XO1AzwkPPE&list=PLU8oAIHdN5BmfwxFO7HdPciOCmmYneAB&index=2>



www.sena.edu.co



**Píldoras
Informáticas**

Cursos de informática

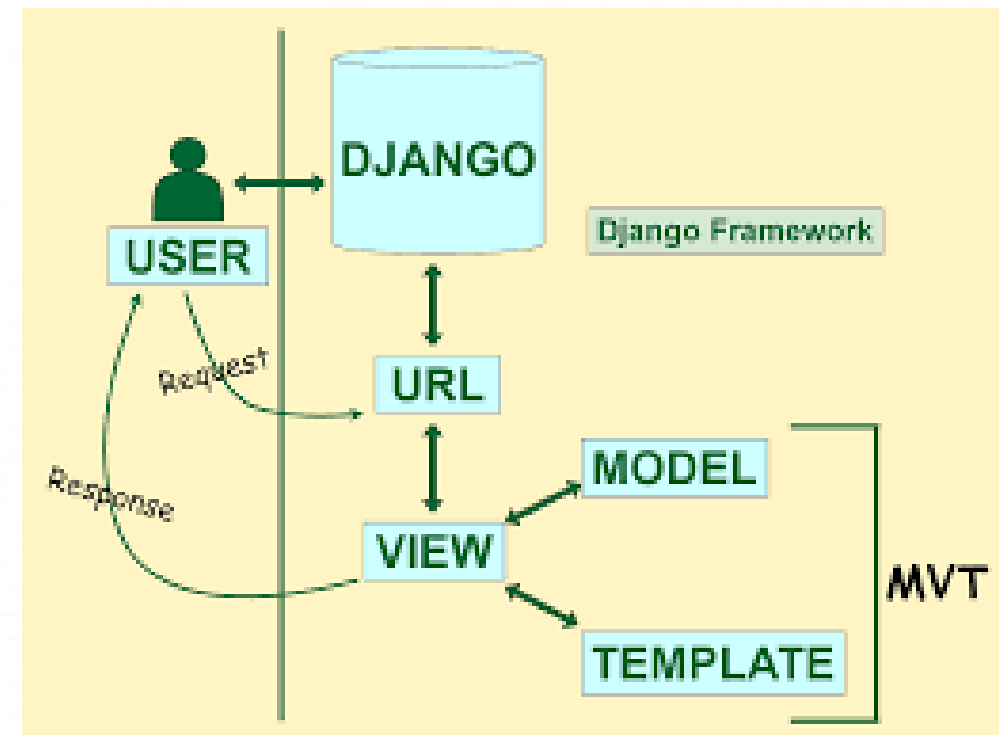
Django Primera Página



1. Creación de Vista
2. Configurar url

Vamos a utilizar dos clases de la biblioteca de Django:

1. Request
2. HttpResponse



Views

Las vistas de Django son funciones de Python que reciben solicitudes http y devuelven una respuesta http, como documentos HTML.

Una página web que utiliza Django está llena de vistas con diferentes tareas y misiones.

Las vistas generalmente se colocan en un archivo llamado `views.py` ubicado en la carpeta de su aplicación.

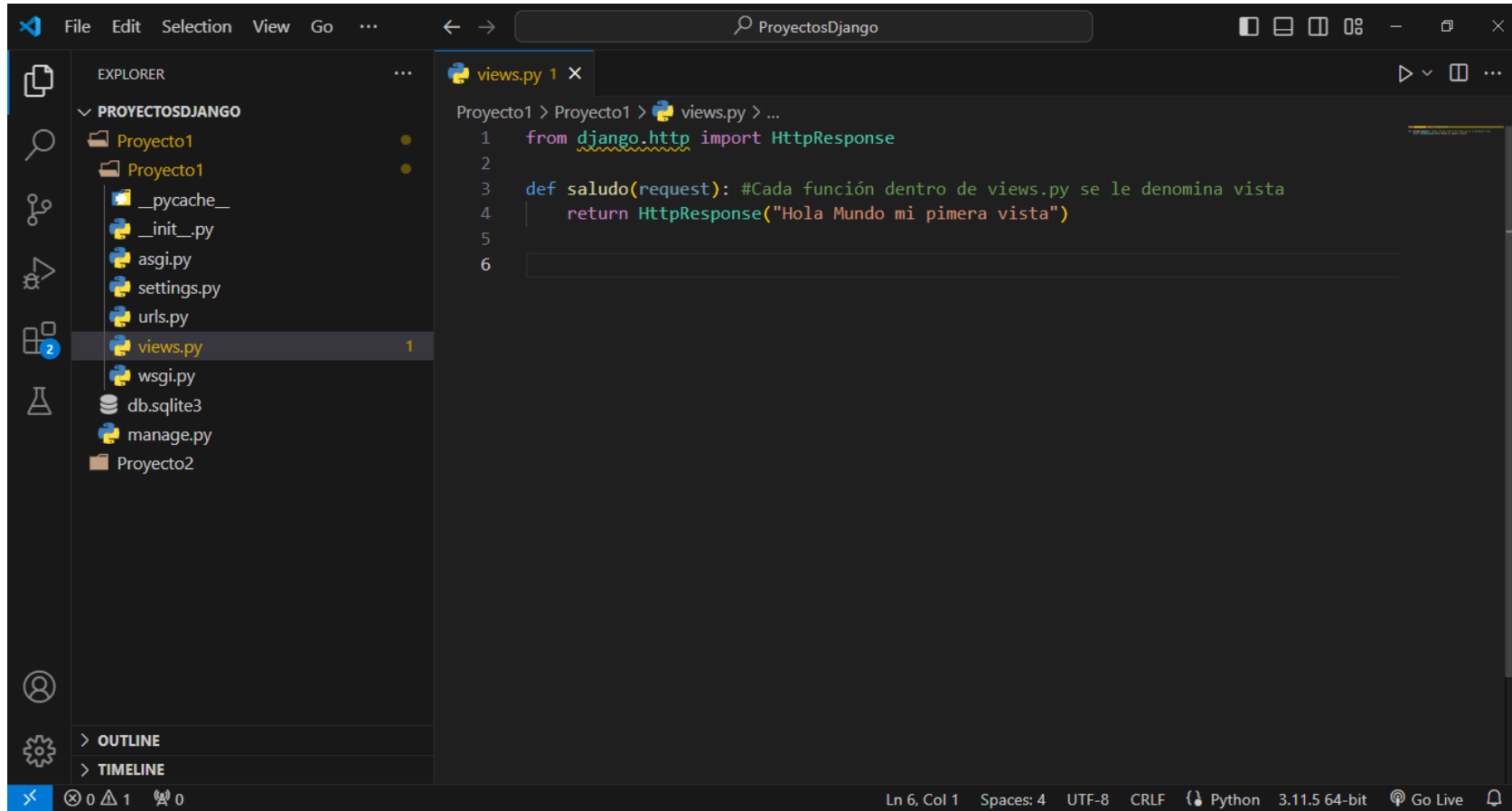
Pasos para crear una vista

1. Define la vista: Define una función de Python en tu archivo `views.py`. Esta función toma un objeto de solicitud como su primer argumento y puede tomar otros argumentos según sea necesario. La vista debe devolver un objeto de respuesta, que generalmente es un objeto `HttpResponse` o un objeto `HttpResponseRedirect` (o alguna variante de estos).
2. Importa los módulos necesarios: Asegúrate de importar los módulos necesarios para trabajar con vistas en Django, como `HttpResponse` desde `django.http`.
3. Escribe la lógica de la vista: Dentro de la función de vista, escribe la lógica necesaria para manejar la solicitud. Esto puede incluir la recuperación de datos de la base de datos, procesamiento de formularios, autenticación de usuarios, etc.

Pasos para crear una vista

4. Retorna una respuesta: Asegúrate de que la vista devuelva una respuesta válida. Esto podría ser un objeto `HttpResponse` que contiene el contenido HTML que desees mostrar, una redirección a otra URL, o cualquier otro tipo de respuesta que necesites.
5. Mapea la URL a la vista: En el archivo **`urls.py`** de tu aplicación, mapea la URL deseada a la función de vista que acabas de crear. Esto se hace utilizando el enrutador de URLs de Django, que normalmente se importa desde `django.urls`.

Pasos para crear una vista



The screenshot shows the Visual Studio Code interface with a Django project named 'ProyectosDjango'. The Explorer panel on the left shows the project structure, with 'views.py' selected. The main editor window displays the code for 'views.py'.

```

ProjectosDjango
├── Proyecto1
│   ├── __pycache__
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   └── views.py
├── Proyecto2
├── db.sqlite3
├── manage.py
└── wsgi.py
  
```

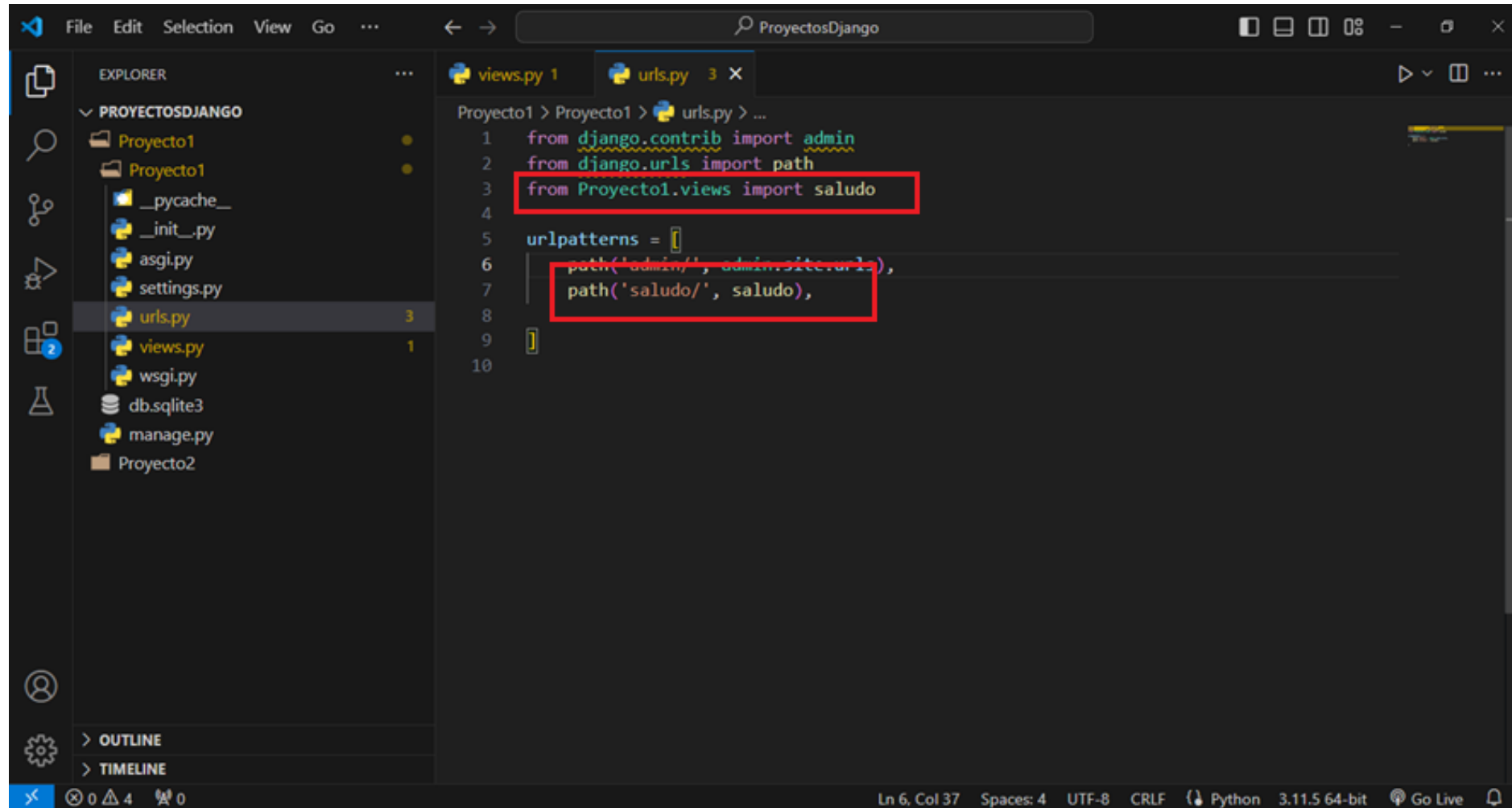
The code in 'views.py' is as follows:

```

1 from django.http import HttpResponse
2
3 def saludo(request): #Cada función dentro de views.py se le denomina vista
4     return HttpResponse("Hola Mundo mi primera vista")
5
6
  
```

The status bar at the bottom indicates the file is at line 6, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.11.5 64-bit interpreter.

Mapeado URL



The screenshot shows a Visual Studio Code editor window with a project named 'ProyectosDjango'. The Explorer sidebar on the left shows the project structure, including 'Proyecto1' and 'Proyecto2'. The 'urls.py' file in 'Proyecto1' is open in the editor. The code in 'urls.py' is as follows:

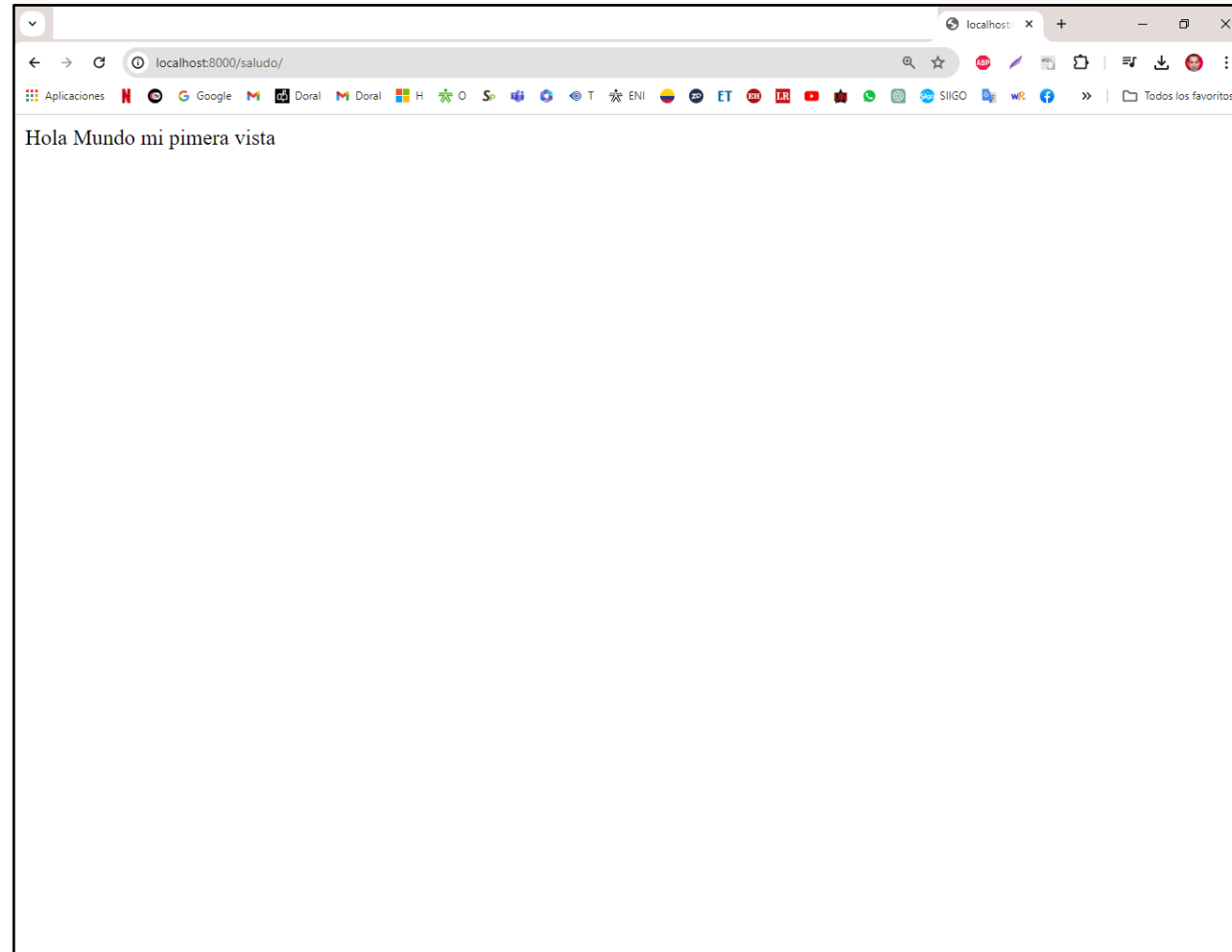
```

1 from django.contrib import admin
2 from django.urls import path
3 from Proyecto1.views import saludo
4
5 urlpatterns = [
6     path('admin/', admin.site.urls),
7     path('saludo/', saludo),
8 ]
9
10

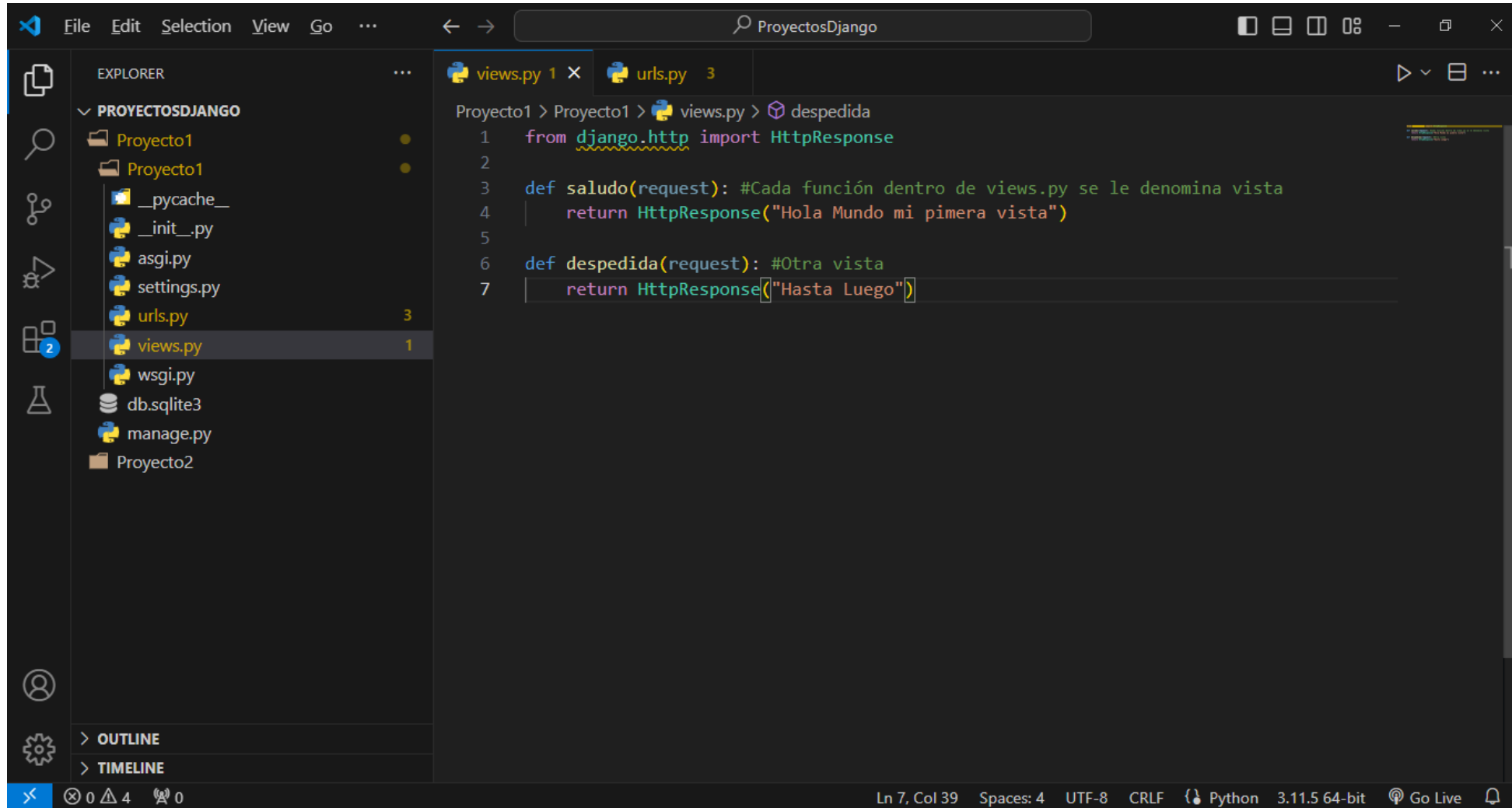
```

Red boxes highlight the import statement on line 3 and the URL pattern on line 7. The status bar at the bottom indicates the file is at line 6, column 37, using UTF-8 encoding and CRLF line endings, with Python 3.11.5 64-bit as the interpreter.

Comprobamos en el Navegador



Creamos una segunda vista



The screenshot shows the Visual Studio Code editor interface for a Django project named 'ProyectosDjango'. The Explorer sidebar on the left displays the project structure, including 'Proyecto1' and 'Proyecto2'. The 'views.py' file is selected and open in the editor. The code in 'views.py' defines two views: 'saludo' and 'despedida'. The 'saludo' view returns a 'Hello World' message, and the 'despedida' view returns a 'Goodbye' message. The status bar at the bottom indicates the current cursor position is at line 7, column 39, with 4 spaces, UTF-8 encoding, and CRLF line endings. The Python version is 3.11.5 64-bit.

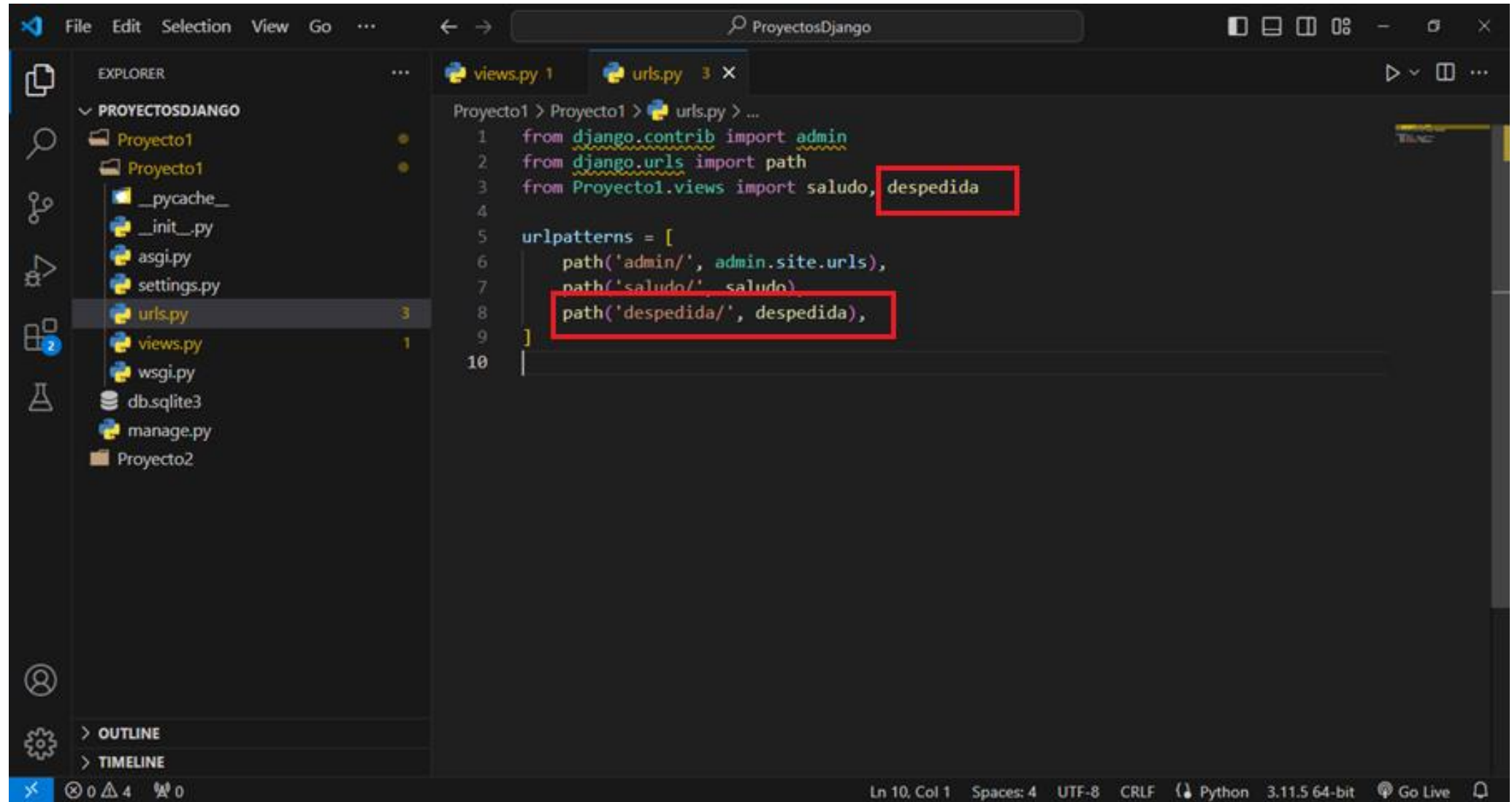
```

ProjectosDjango
EXPLORER
PROJECTOSDJANGO
  Proyecto1
    Proyecto1
      __pycache__
      __init__.py
      asgi.py
      settings.py
      urls.py 3
      views.py 1
      wsgi.py
      db.sqlite3
      manage.py
      Proyecto2
  OUTLINE
  TIMELINE

ProjectosDjango > Proyecto1 > Proyecto1 > views.py > despedida
1  from django.http import HttpResponse
2
3  def saludo(request): #Cada función dentro de views.py se le denomina vista
4      return HttpResponse("Hola Mundo mi pimera vista")
5
6  def despedida(request): #Otra vista
7      return HttpResponse("Hasta Luego")

```

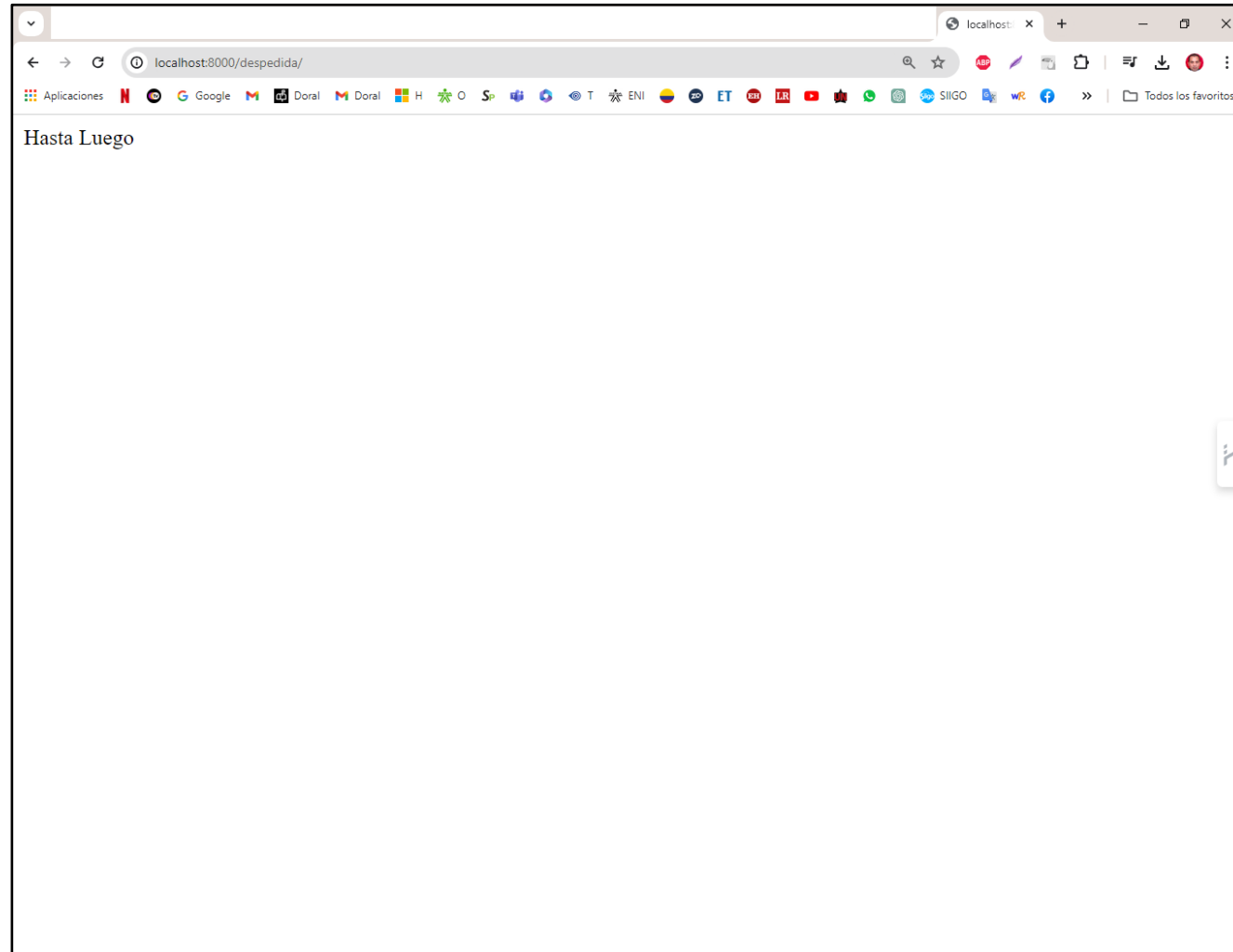
Mapeado URL



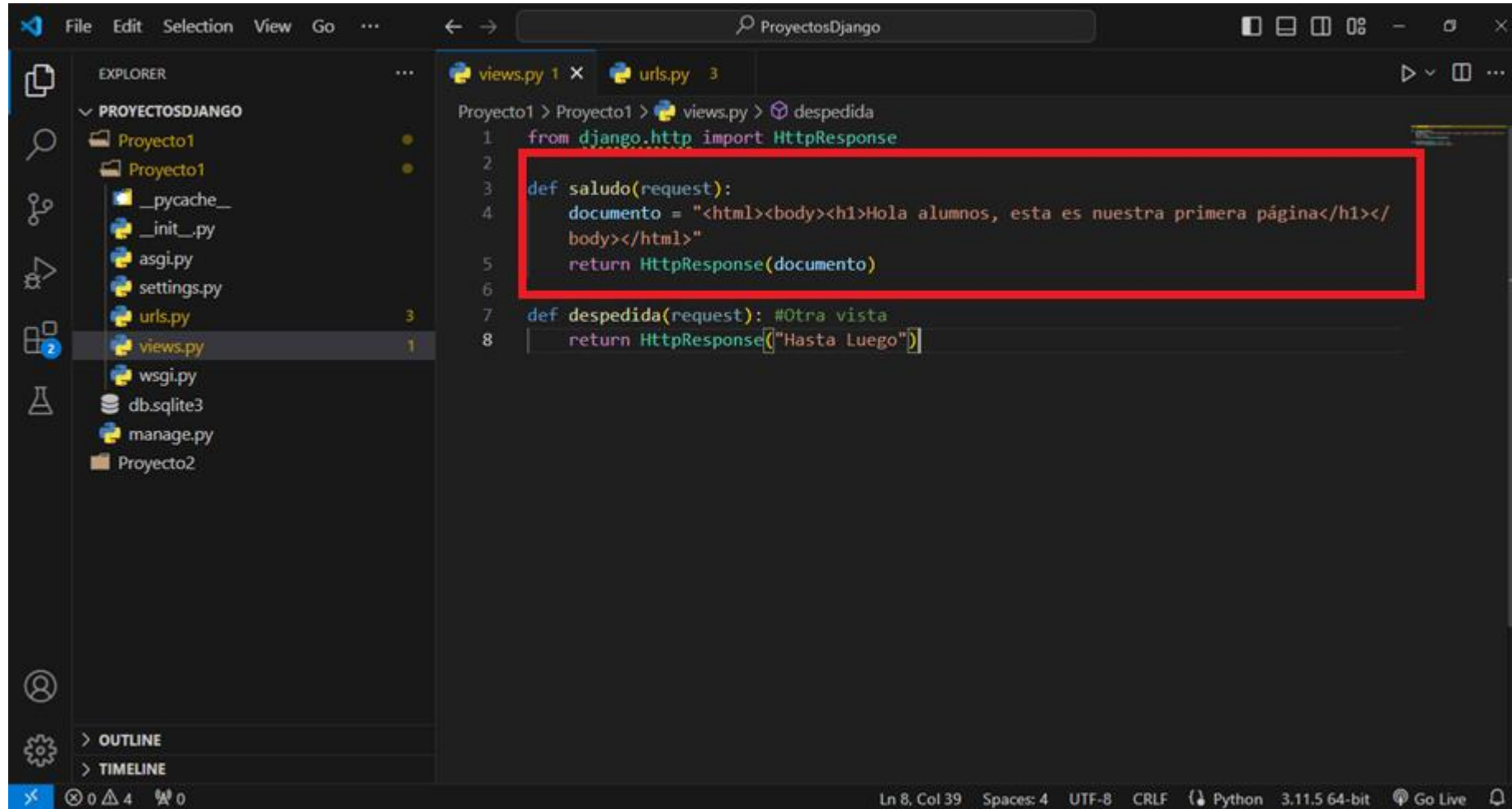
The screenshot shows the Visual Studio Code editor with a Django project named 'ProyectosDjango'. The Explorer sidebar on the left shows the project structure, with 'urls.py' selected under 'Proyecto1'. The main editor window displays the content of 'urls.py', which includes imports for 'admin', 'path', and 'saludo' from 'Proyecto1.views'. The 'urlpatterns' list contains three entries: 'admin/', 'saludo/' (mapped to 'saludo'), and 'despedida/' (mapped to 'despedida'). The 'despedida' view name and its corresponding path are highlighted with red boxes. The status bar at the bottom indicates the file is at line 10, column 1, using UTF-8 encoding with CRLF line endings, and is being edited in Python 3.11.5 64-bit.

```
1 from django.contrib import admin
2 from django.urls import path
3 from Proyecto1.views import saludo, despedida
4
5 urlpatterns = [
6     path('admin/', admin.site.urls),
7     path('saludo/', saludo),
8     path('despedida/', despedida),
9 ]
10
```

Comprobamos en el Navegador



Incluir html en nuestras views



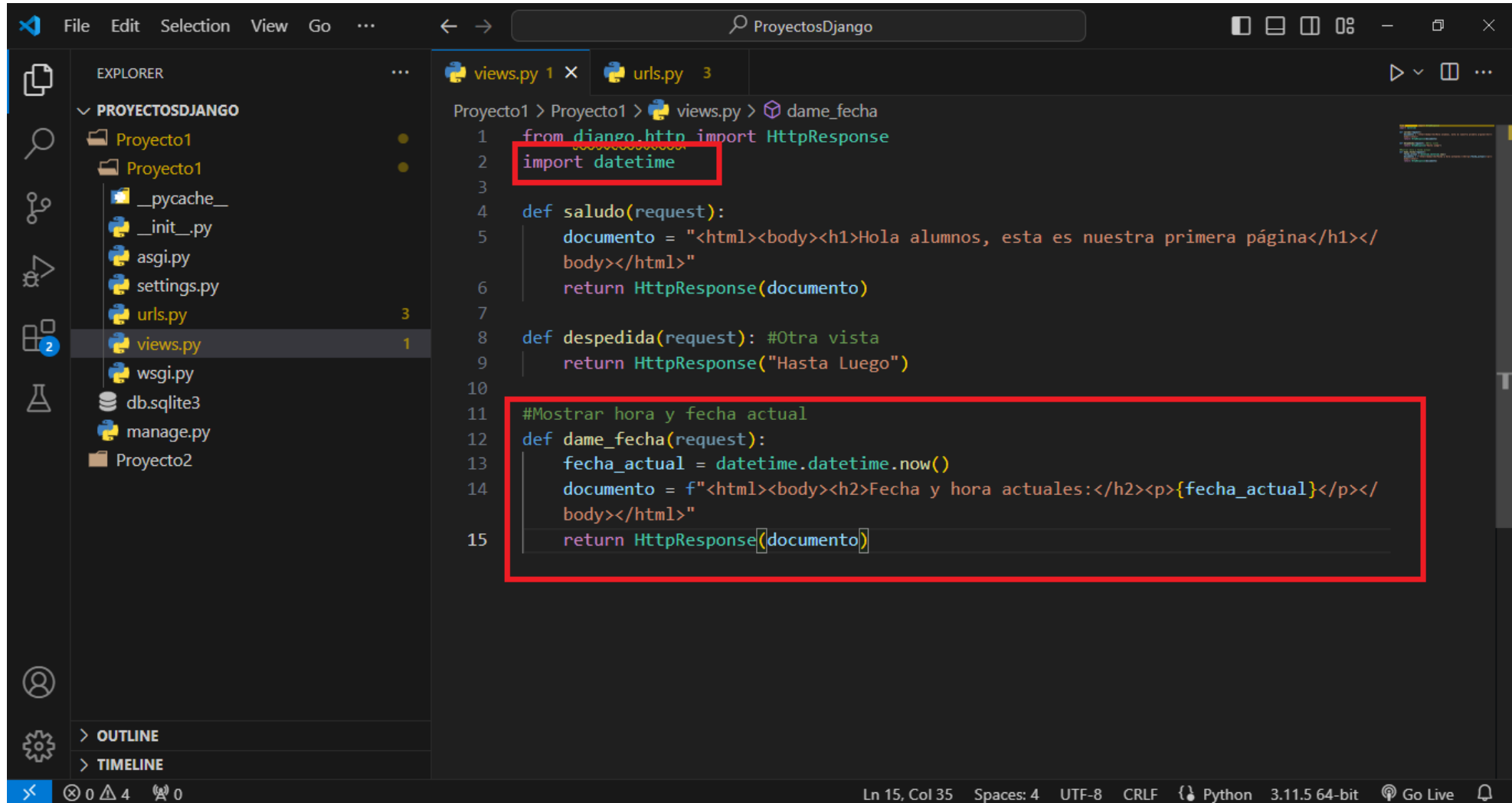
The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the project structure for 'PROYECTOSDJANGO', including 'Proyecto1' and 'Proyecto2'. The 'views.py' file is selected and highlighted. The main editor window shows the code for 'views.py'. A red rectangle highlights the following code block:

```

1  from django.http import HttpResponse
2
3  def saludo(request):
4      documento = "<html><body><h1>Hola alumnos, esta es nuestra primera página</h1></body></html>"
5      return HttpResponse(documento)
6
7  def despedida(request): #Otra vista
8      return HttpResponse("Hasta Luego")
    
```

The status bar at the bottom indicates the current position is Line 8, Column 39, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.11.5 64-bit, and the Go Live extension is active.

Mostrar hora y fecha actual



```
File Edit Selection View Go ... < > ProyectosDjango
```

EXPLORER

- PROYECTOSDJANGO
 - Proyecto1
 - Proyecto1
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py 3
 - views.py 1
 - wsgi.py
 - db.sqlite3
 - manage.py
 - Proyecto2

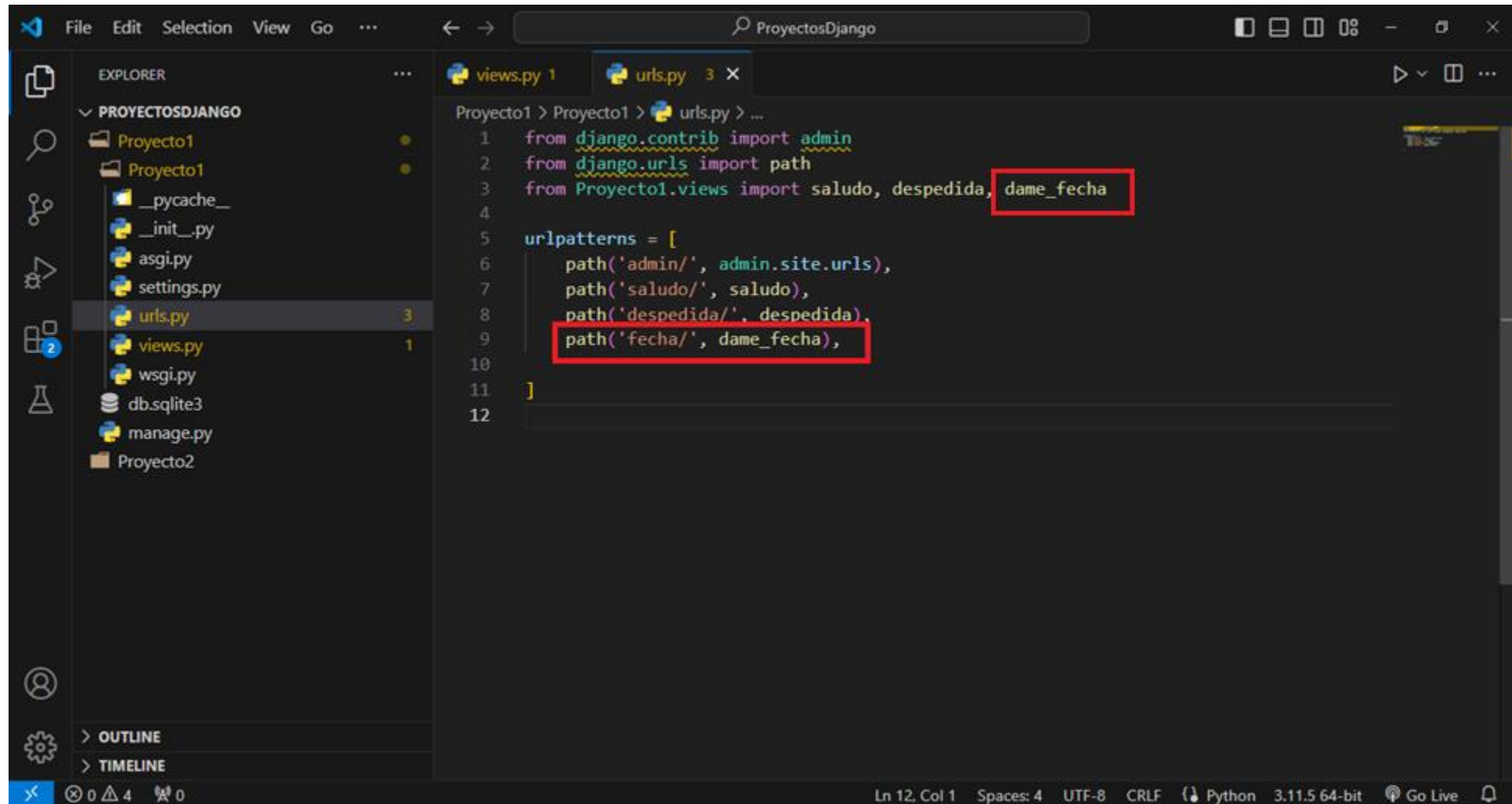
views.py 1 x urls.py 3

Proyecto1 > Proyecto1 > views.py > dame_fecha

```
1 from django.http import HttpResponse
2 import datetime
3
4 def saludo(request):
5     documento = "<html><body><h1>Hola alumnos, esta es nuestra primera página</h1></body></html>"
6     return HttpResponse(documento)
7
8 def despedida(request): #Otra vista
9     return HttpResponse("Hasta Luego")
10
11 #Mostrar hora y fecha actual
12 def dame_fecha(request):
13     fecha_actual = datetime.datetime.now()
14     documento = f"<html><body><h2>Fecha y hora actuales:</h2><p>{fecha_actual}</p></body></html>"
15     return HttpResponse(documento)
```

Ln 15, Col 35 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live

Mapeado URL



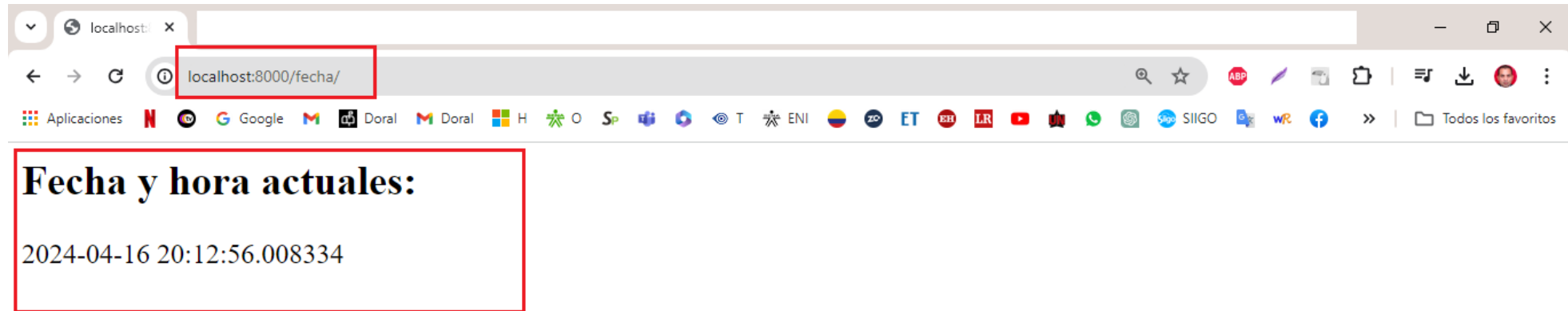
The screenshot shows the Visual Studio Code editor with a Django project named 'ProyectosDjango'. The Explorer panel on the left shows the project structure, with 'urls.py' selected under 'Proyecto1'. The main editor window displays the content of 'urls.py', which includes imports for 'admin', 'path', and 'saludo', 'despedida', and 'dame_fecha' from 'Proyecto1.views'. The 'urlpatterns' list contains three path entries: 'admin/', 'saludo/', and 'fecha/'. The 'dame_fecha' import and the 'path('fecha/', dame_fecha)' entry are highlighted with red boxes.

```

1  from django.contrib import admin
2  from django.urls import path
3  from Proyecto1.views import saludo, despedida, dame_fecha
4
5  urlpatterns = [
6      path('admin/', admin.site.urls),
7      path('saludo/', saludo),
8      path('despedida/', despedida),
9      path('fecha/', dame_fecha),
10 ]
11
12

```

Comprobamos en el Navegador

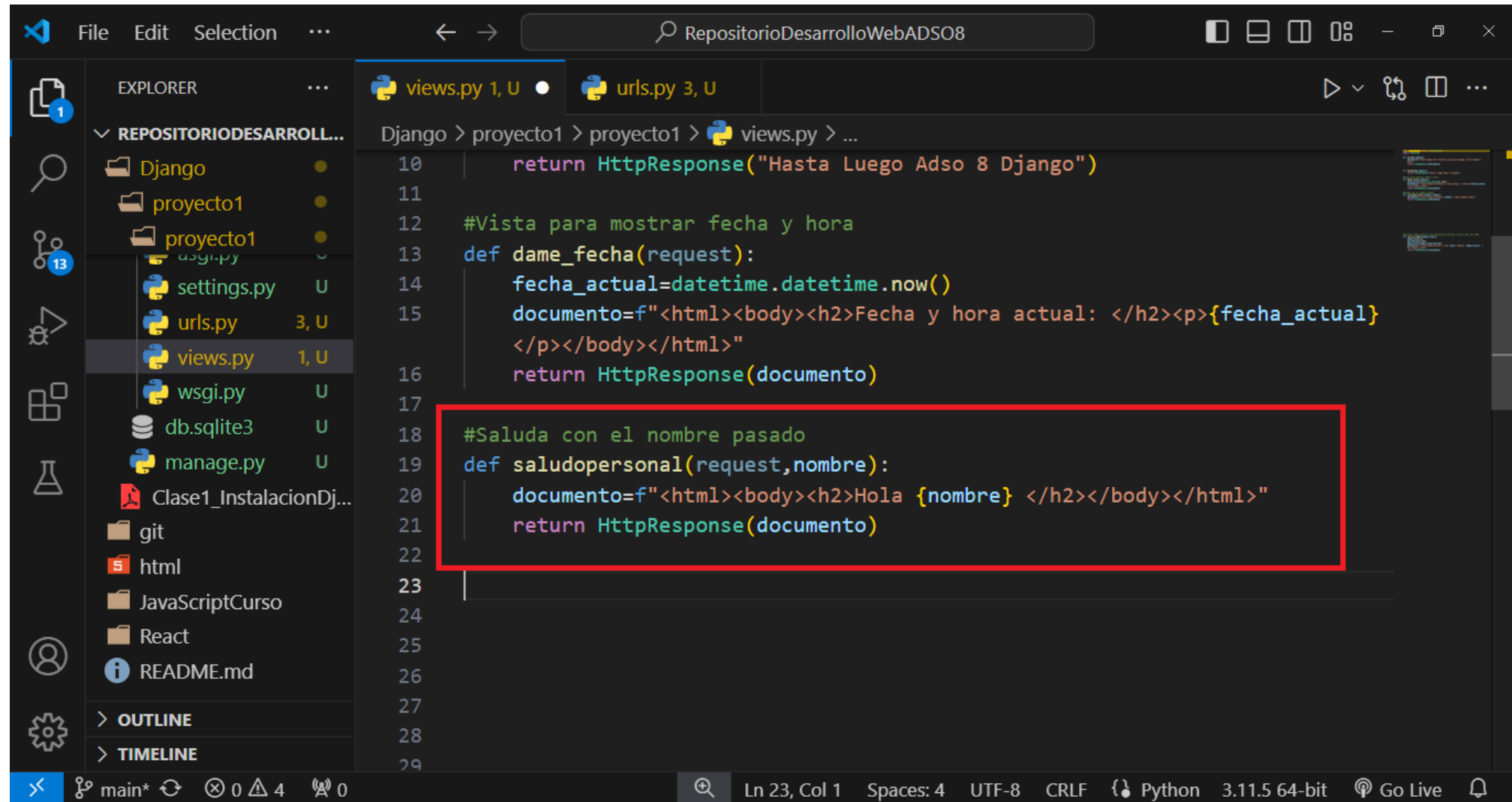


Parámetros en URL

El paso de parámetros en las URL es una característica crucial en el desarrollo web, especialmente en el contexto de Django. Permite a los desarrolladores crear aplicaciones web dinámicas y personalizadas al pasar información específica a través de la URL. Esto facilita la creación de páginas que responden a diferentes contextos o necesidades del usuario.

Importancia del paso de parámetros en URL:

- ✓ Personalización de la experiencia del usuario
- ✓ Facilita la navegación y la compartición de enlaces



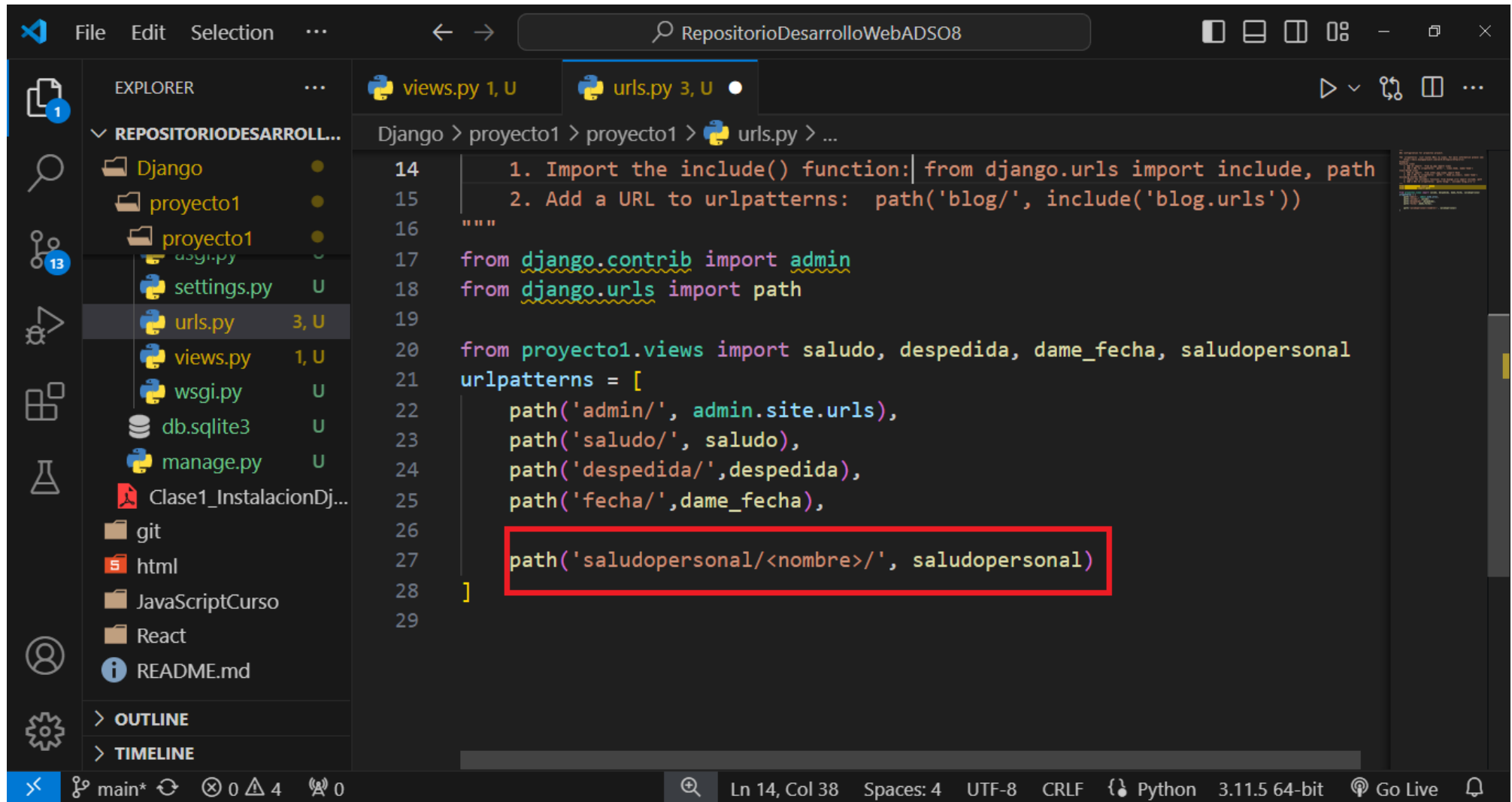
The screenshot shows the Visual Studio Code interface with a Django project named 'RepositorioDesarrolloWebADSO8'. The Explorer panel on the left shows the project structure, including 'Django', 'proyecto1', and various Python files. The main editor window displays the 'views.py' file, which contains two view functions: 'dame_fecha' and 'saludopersonal'. The 'saludopersonal' function is highlighted with a red box, indicating it is the focus of the current task. The status bar at the bottom shows the current cursor position at line 23, column 1, and the file encoding as UTF-8.

```

10     return HttpResponse("Hasta Luego Adso 8 Django")
11
12     #Vista para mostrar fecha y hora
13     def dame_fecha(request):
14         fecha_actual=datetime.datetime.now()
15         documento=f"<html><body><h2>Fecha y hora actual: </h2><p>{fecha_actual}</p></body></html>"
16         return HttpResponse(documento)
17
18     #Saluda con el nombre pasado
19     def saludopersonal(request,nombre):
20         documento=f"<html><body><h2>Hola {nombre} </h2></body></html>"
21         return HttpResponse(documento)
22
23
24
25
26
27
28
29

```

Mapeado URL



The screenshot shows a Visual Studio Code editor window with the Django project structure on the left and the `urls.py` file open in the editor. The file contains the following code:

```

14 1. Import the include() function: from django.urls import include, path
15 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path
19
20 from proyecto1.views import saludo, despedida, dame_fecha, saludopersonal
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('saludo/', saludo),
24     path('despedida/', despedida),
25     path('fecha/', dame_fecha),
26
27     path('saludopersonal/<nombre>/', saludopersonal)
28 ]
29

```

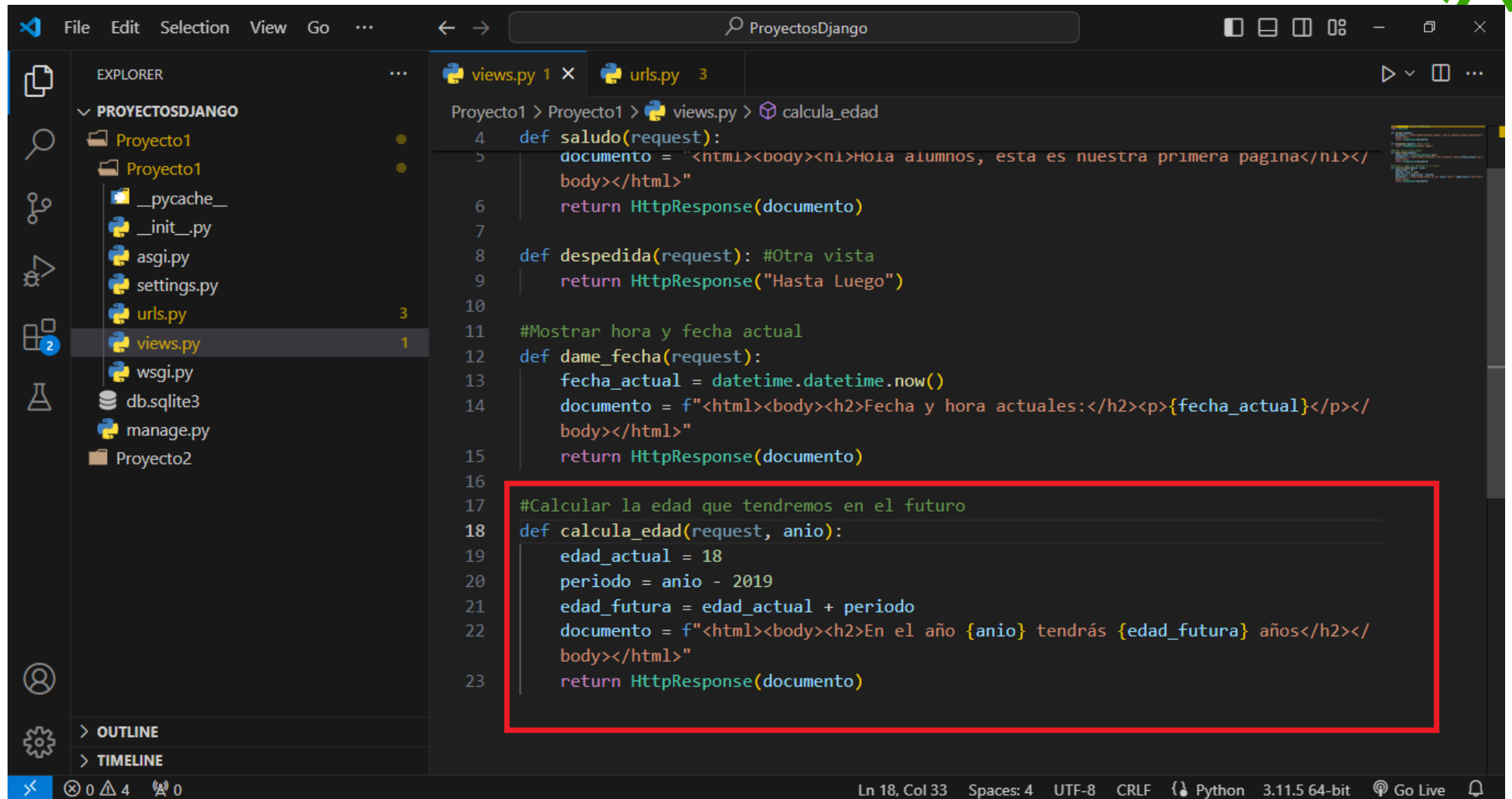
The last line of the `urlpatterns` list is highlighted with a red box:

```

path('saludopersonal/<nombre>/', saludopersonal)

```

The status bar at the bottom indicates the current file is `urls.py` at line 14, column 38, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.11.5 64-bit.

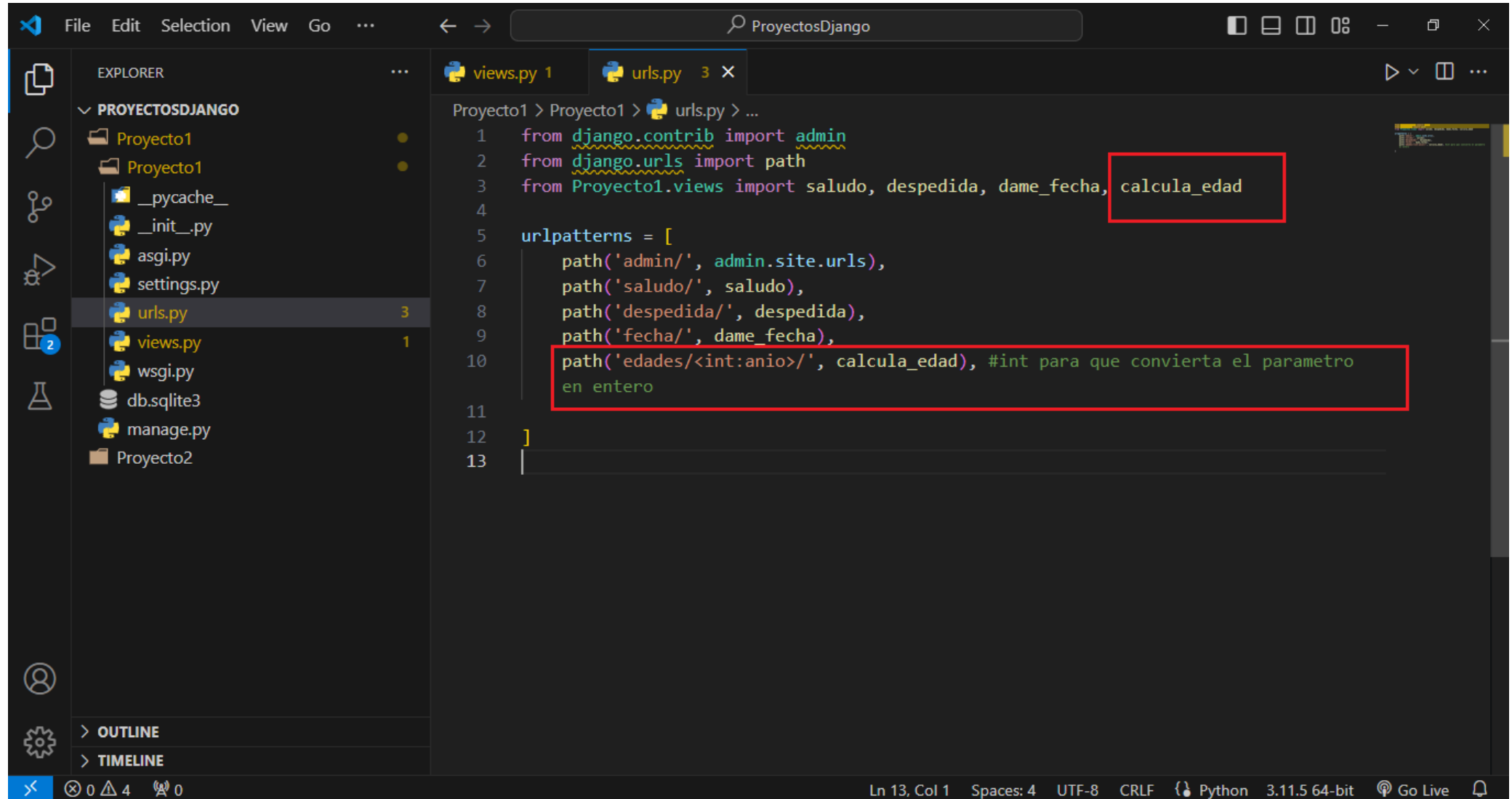


The screenshot shows a VS Code editor window with a Django project named 'ProyectosDjango'. The Explorer panel on the left shows the project structure, including 'Proyecto1' and 'Proyecto2'. The 'views.py' file is open in the editor, showing a function 'calcula_edad' that calculates the future age based on the current year (2019) and a specified number of years ('anio'). The function returns an HTML response with the calculated age.

```

4 def saludo(request):
5     documento = "<html><body><n1>Hola alumnos, esta es nuestra primera pagina</n1></
6     body></html>"
7     return HttpResponse(documento)
8
9 def despedida(request): #Otra vista
10     return HttpResponse("Hasta Luego")
11
12 #Mostrar hora y fecha actual
13 def dame_fecha(request):
14     fecha_actual = datetime.datetime.now()
15     documento = f"<html><body><h2>Fecha y hora actuales:</h2><p>{fecha_actual}</p></
16     body></html>"
17     return HttpResponse(documento)
18
19 #Calcular la edad que tendremos en el futuro
20 def calcula_edad(request, anio):
21     edad_actual = 18
22     periodo = anio - 2019
23     edad_futura = edad_actual + periodo
24     documento = f"<html><body><h2>En el año {anio} tendrás {edad_futura} años</h2></
25     body></html>"
26     return HttpResponse(documento)
  
```

Mapeado URL



The screenshot shows the Visual Studio Code editor with a project named 'ProyectosDjango'. The Explorer panel on the left shows the project structure, with 'urls.py' selected under 'Proyecto1'. The main editor window displays the content of 'urls.py'.

```

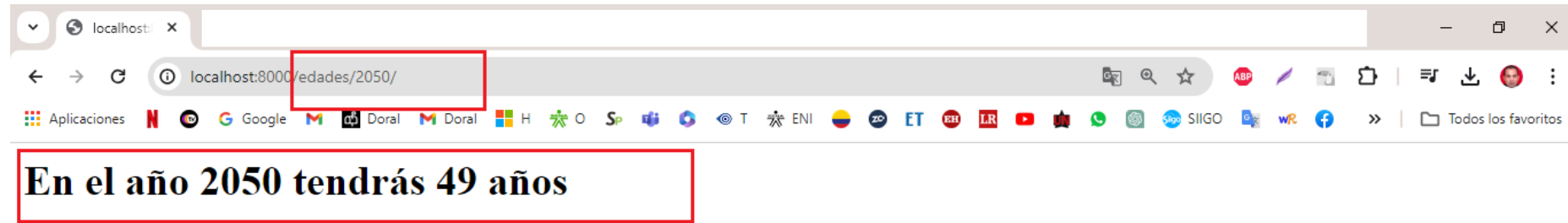
1  from django.contrib import admin
2  from django.urls import path
3  from Proyecto1.views import saludo, despedida, dame_fecha, calcula_edad
4
5  urlpatterns = [
6      path('admin/', admin.site.urls),
7      path('saludo/', saludo),
8      path('despedida/', despedida),
9      path('fecha/', dame_fecha),
10     path('edades/<int:anio>/', calcula_edad), #int para que convierta el parametro
11                                           en entero
12 ]
13

```

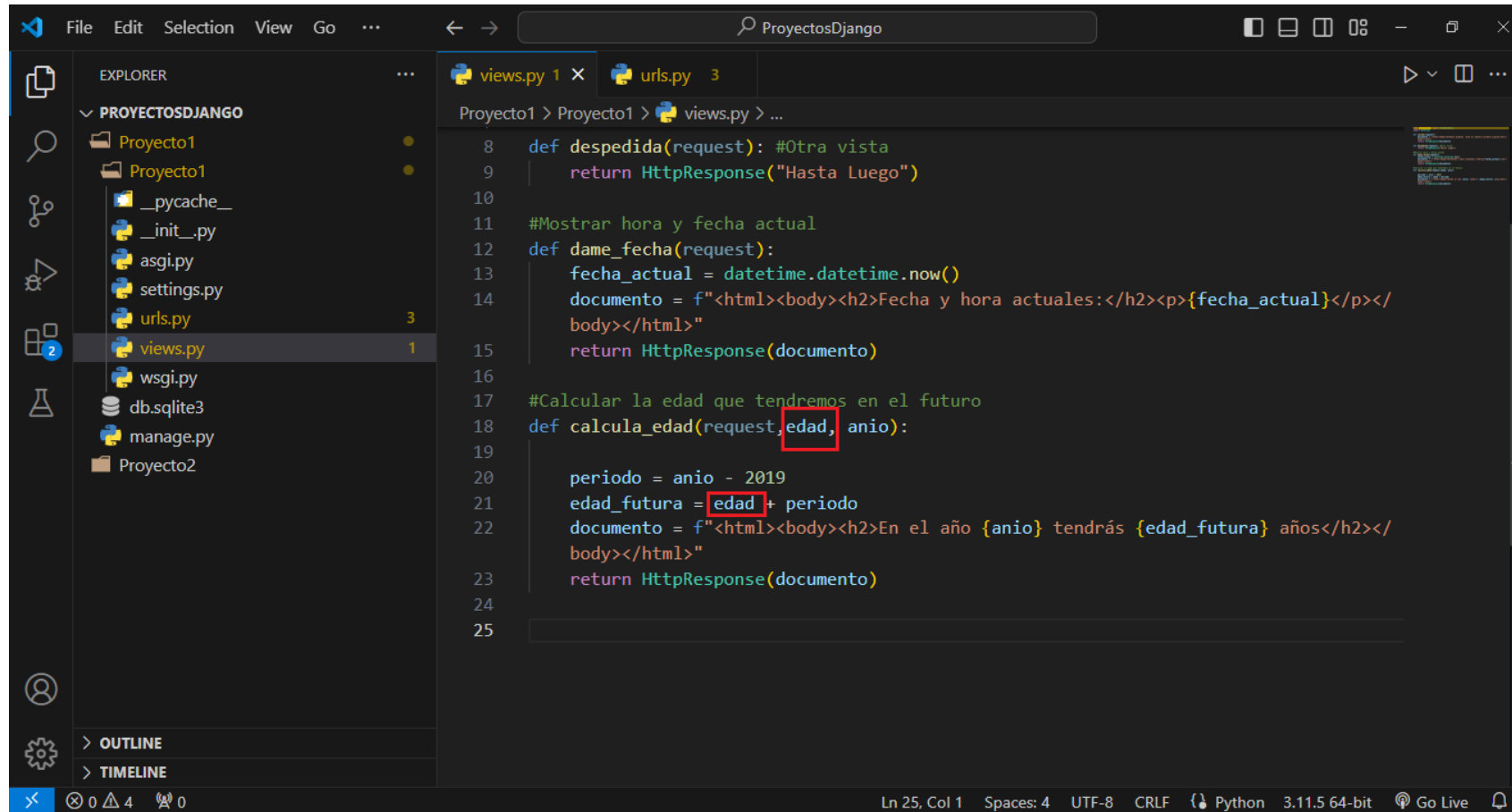
Two red boxes highlight specific parts of the code: the first box highlights the import of `calcula_edad` from `Proyecto1.views` on line 3, and the second box highlights the URL pattern `path('edades/<int:anio>/', calcula_edad)` on line 10, along with its comment.

The status bar at the bottom indicates the current position is Line 13, Column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.11.5 64-bit, and the Go Live extension is active.

Comprobamos en el Navegador



Calcular edad futura pasando la edad actual

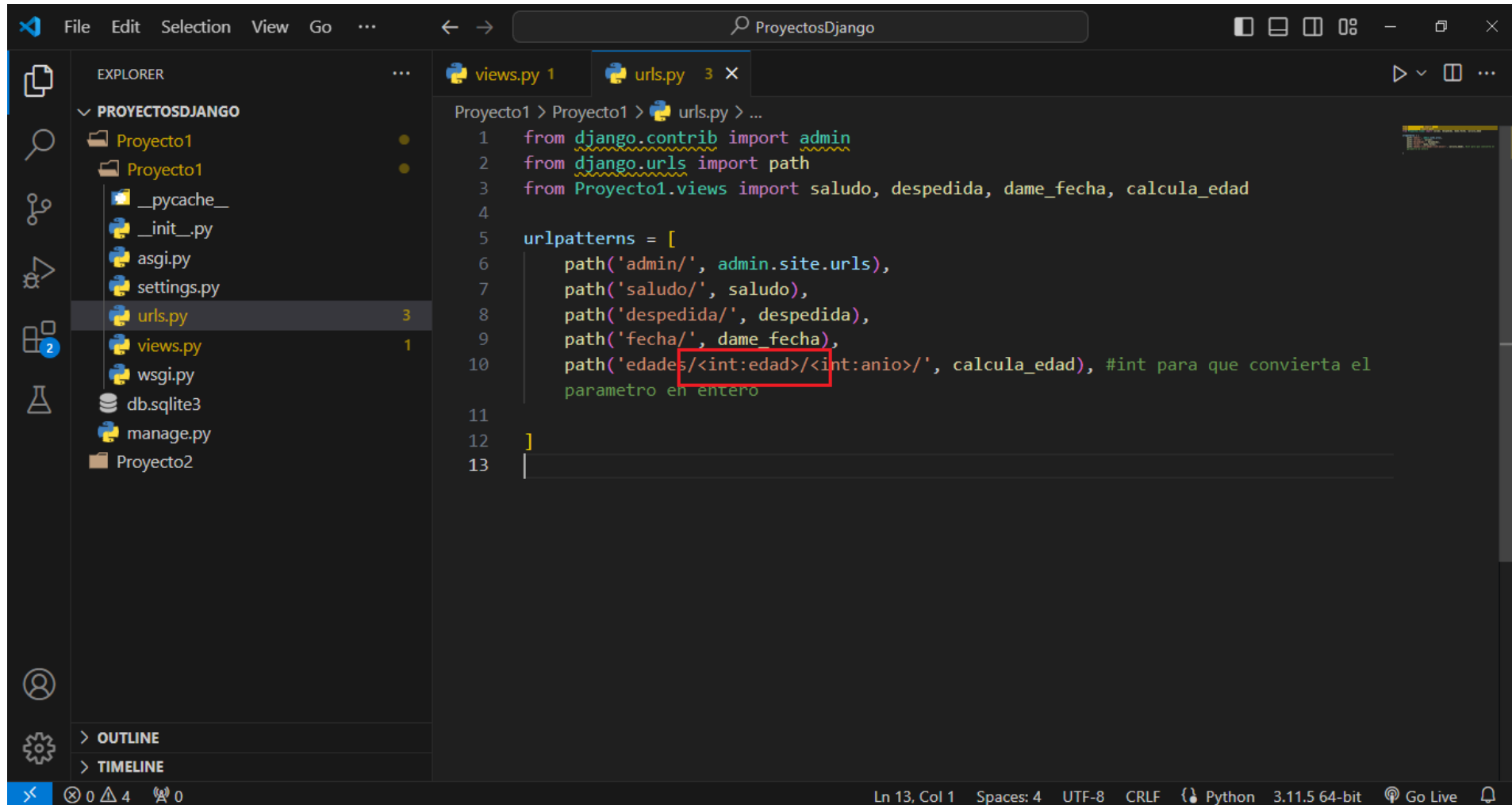
The screenshot shows a Visual Studio Code editor window with a Django project named 'ProyectosDjango'. The Explorer sidebar on the left shows the project structure, including 'Proyecto1' and 'Proyecto2'. The 'views.py' file is selected and open in the editor. The code in 'views.py' includes a function 'despedida' and a function 'dame_fecha'. A new function 'calcula_edad' is being defined, which takes 'request', 'edad', and 'anio' as arguments. The 'edad' parameter is highlighted with a red box. The function calculates the future age by subtracting the year 2019 from the 'anio' and adding the 'edad' to the result. The 'edad' variable is also highlighted with a red box in the calculation.

```

8 def despedida(request): #Otra vista
9     return HttpResponse("Hasta Luego")
10
11 #Mostrar hora y fecha actual
12 def dame_fecha(request):
13     fecha_actual = datetime.datetime.now()
14     documento = f"<html><body><h2>Fecha y hora actuales:</h2><p>{fecha_actual}</p></body></html>"
15     return HttpResponse(documento)
16
17 #Calcular la edad que tendremos en el futuro
18 def calcula_edad(request, edad, anio):
19
20     periodo = anio - 2019
21     edad_futura = edad + periodo
22     documento = f"<html><body><h2>En el año {anio} tendrás {edad_futura} años</h2></body></html>"
23     return HttpResponse(documento)
24
25

```

Mapeado URL

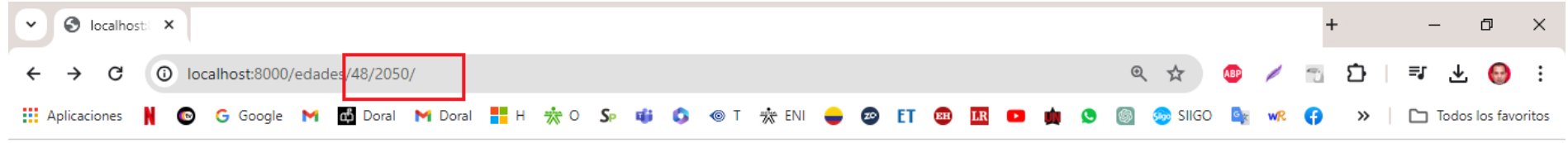


The screenshot shows the Visual Studio Code editor with a Django project named 'ProyectosDjango'. The Explorer sidebar on the left shows the project structure, with 'urls.py' selected under 'Proyecto1'. The main editor window displays the content of 'urls.py', which includes imports for Django's admin, path, and project-specific views, followed by a list of URL patterns. A red rectangle highlights the regex pattern '<int:edad>/' in the 'edades' path, with a comment explaining its purpose.

```
1 from django.contrib import admin
2 from django.urls import path
3 from Proyecto1.views import saludo, despedida, dame_fecha, calcula_edad
4
5 urlpatterns = [
6     path('admin/', admin.site.urls),
7     path('saludo/', saludo),
8     path('despedida/', despedida),
9     path('fecha/', dame_fecha),
10    path('edades/<int:edad>/', calcula_edad), #int para que convierta el
11                                           parametro en entero
12 ]
13
```

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live

Comprobamos en el Navegador



En el año 2050 tendrás 79 años



Ejercicios para desarrollar

1. Crea una vista llamada "saludo_personalizado" que tome un parámetro en la URL llamado "nombre" y devuelva un mensaje de saludo personalizado con ese nombre.
2. Crea una vista llamada "perfil_usuario" que tome dos parámetros en la URL, "usuario" y "seccion", y devuelva un mensaje que muestre el perfil de usuario y la sección solicitada.
3. Enunciado: Crea una vista llamada "calcular_hipotenusa" que tome dos parámetros en la URL, "lado_a" y "lado_b", que representen los catetos de un triángulo rectángulo. La vista debe calcular la hipotenusa del triángulo y devolver el resultado. Validar en la vista que los valores recibidos sean numéricos.
4. Crea una vista llamada "mayor_edad" que tome un parámetro en la URL llamado "edad" y devuelva un mensaje indicando si la persona es mayor o menor de edad.

Ejercicios para desarrollar

- Una empresa desea calcular el salario mensual de los empleados cuyo trabajo se paga en función de las horas laboradas y el valor de la hora. A este salario se le deberá descontar el valor por concepto de impuestos en un porcentaje indicado por el empleador.

Crea una vista llamada "salario_empleado" que tome los parámetros en la URL nombre del empleado, cantidad de horas trabajadas en el mes, valor base de la hora, porcentaje de impuesto a descontar y devuelva un mensaje proporcione al empleador la información del salario a pagar al empleado. La cadena de salida deberá indicar lo siguiente: "El salario después de impuestos del empleado {nombre} es: \${salario_neto}".

Nombre	Horas Trabajadas	Valor Base Hora	Porcentaje Impuesto	return
Luis Iglesias	160	25000	20	El salario después de impuestos del empleado Luis Iglesias es: \$3200000.0



GRACIAS

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682



www.sena.edu.co