
Eva3

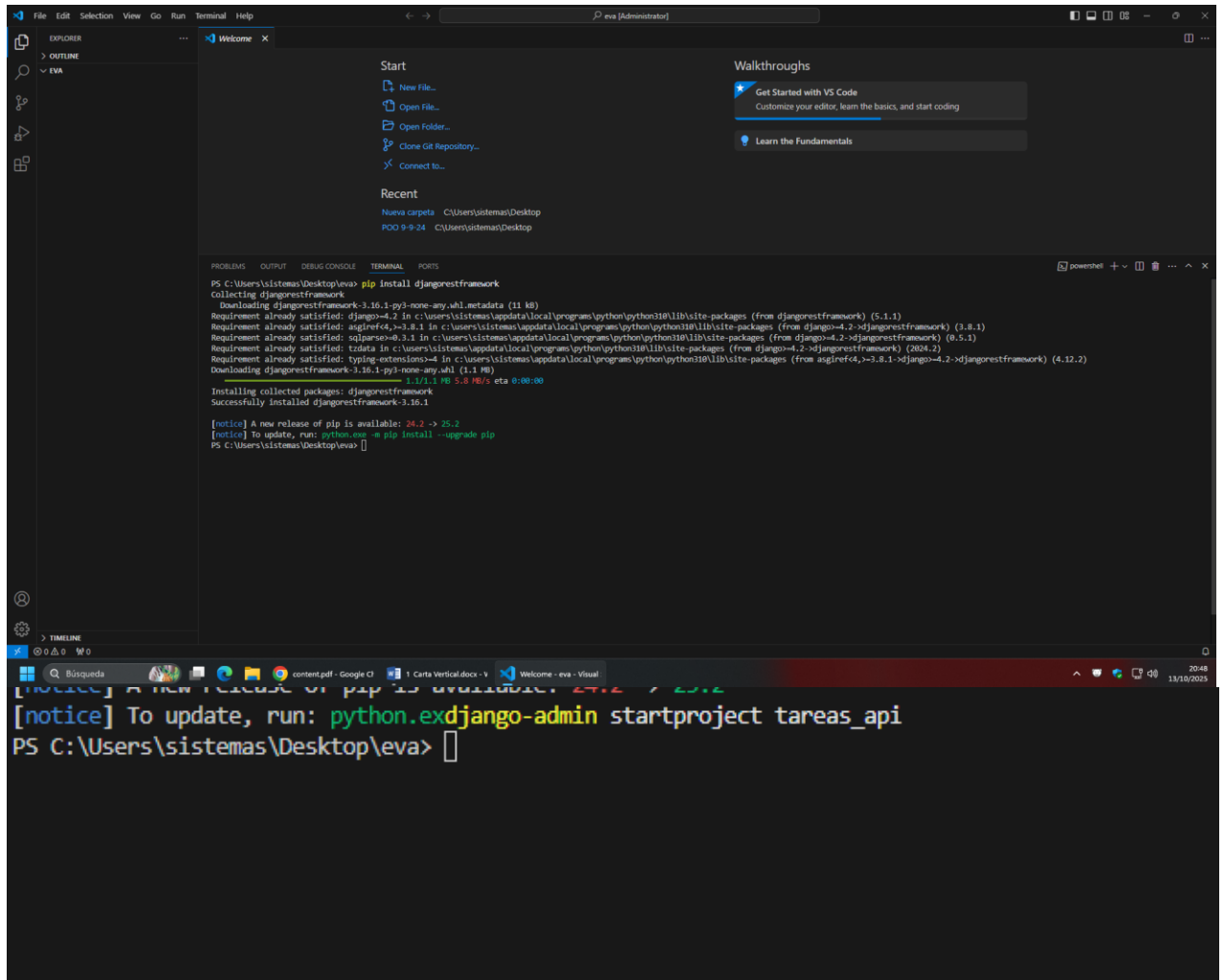
NOMBRE: Pedro Romo

CARRERA: Ingeniería informática

ASIGNATURA: Programación Back End

PROFESOR: Miguel Angel Chacana González

FECHA: 13/10/2025

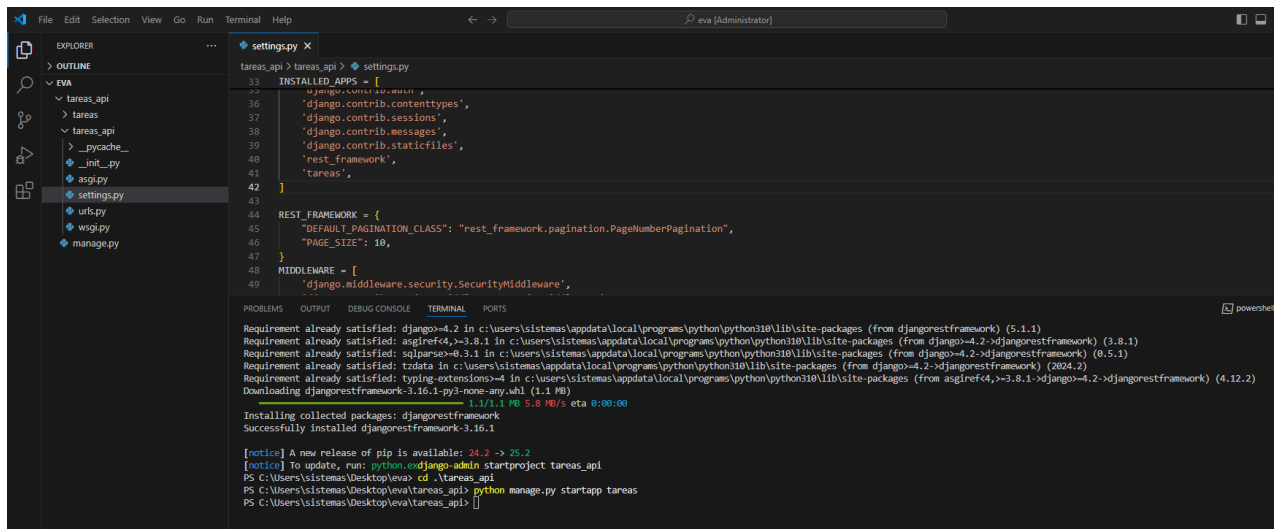


```

PS C:\Users\sistemas\Desktop\eva> pip install djangorestframework
Collecting djangorestframework
  Downloading djangorestframework-3.16.1-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: django>4.2 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from djangorestframework) (5.1.1)
Requirement already satisfied: asgiref<4, >=3.8.1 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from django>4.2->djangorestframework) (3.8.1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from django>4.2->djangorestframework) (0.5.1)
Requirement already satisfied: tzdata in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from django>4.2->djangorestframework) (2024.2)
Requirement already satisfied: typing-extensions>=4 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from asgiref<4, >=3.8.1->django>4.2->djangorestframework) (4.12.2)
Downloading djangorestframework-3.16.1-py3-none-any.whl (1.1 MB)
  1.1/1.1 MB 5.8 MB/s eta 0:00:00
Installing collected packages: djangorestframework
Successfully installed djangorestframework-3.16.1

[notice] A new release of pip is available: 24.2 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\sistemas\Desktop\eva>

```



```

tareas_api > tareas_api > settings.py
33 INSTALLED_APPS = [
34     'django.contrib.auth',
35     'django.contrib.contenttypes',
36     'django.contrib.sessions',
37     'django.contrib.messages',
38     'django.contrib.staticfiles',
39     'rest_framework',
40     'tareas',
41 ]
42
43
44 REST_FRAMEWORK = {
45     'DEFAULT_PAGINATION_CLASS': 'rest_framework.pagination.PageNumberPagination',
46     'PAGE_SIZE': 10,
47 }
48
49 MIDDLEWARE = [
50     'django.middleware.security.SecurityMiddleware',
51 ]
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

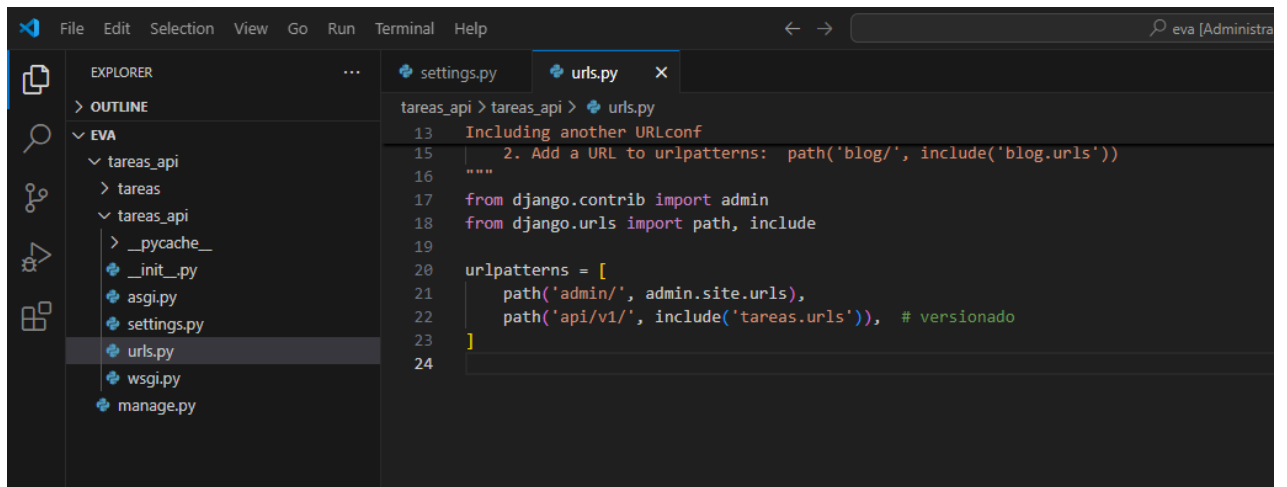
```

```

Requirement already satisfied: django>4.2 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from djangorestframework) (5.1.1)
Requirement already satisfied: asgiref<4, >=3.8.1 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from django>4.2->djangorestframework) (3.8.1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from django>4.2->djangorestframework) (0.5.1)
Requirement already satisfied: tzdata in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from django>4.2->djangorestframework) (2024.2)
Requirement already satisfied: typing-extensions>=4 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from asgiref<4, >=3.8.1->django>4.2->djangorestframework) (4.12.2)
Downloading djangorestframework-3.16.1-py3-none-any.whl (1.1 MB)
  1.1/1.1 MB 5.8 MB/s eta 0:00:00
Installing collected packages: djangorestframework
Successfully installed djangorestframework-3.16.1

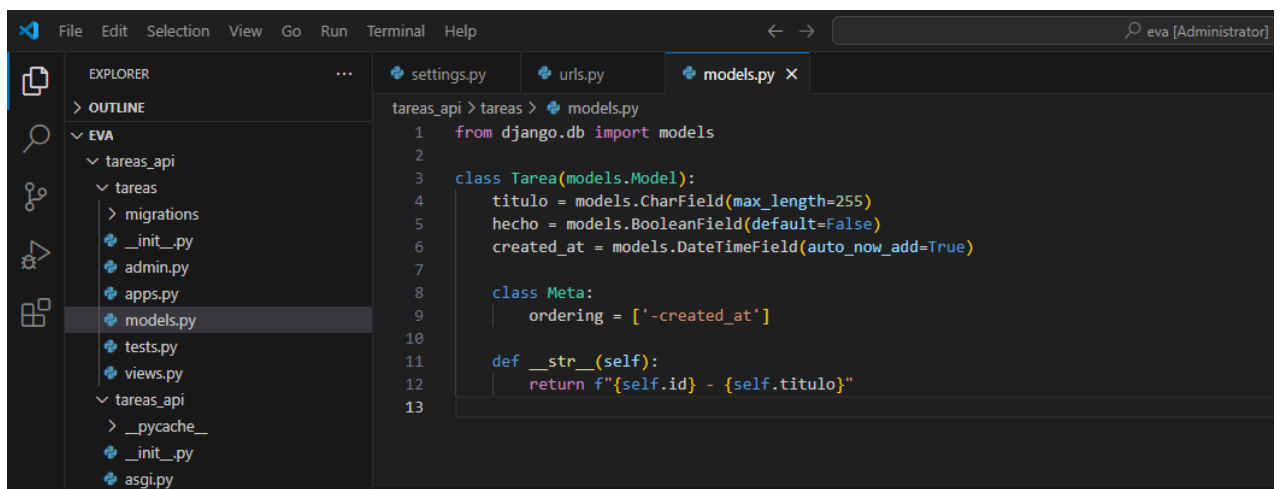
[notice] A new release of pip is available: 24.2 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\sistemas\Desktop\eva> cd .\tareas_api
PS C:\Users\sistemas\Desktop\eva\tareas_api> python manage.py startproject tareas_api
PS C:\Users\sistemas\Desktop\eva\tareas_api>

```



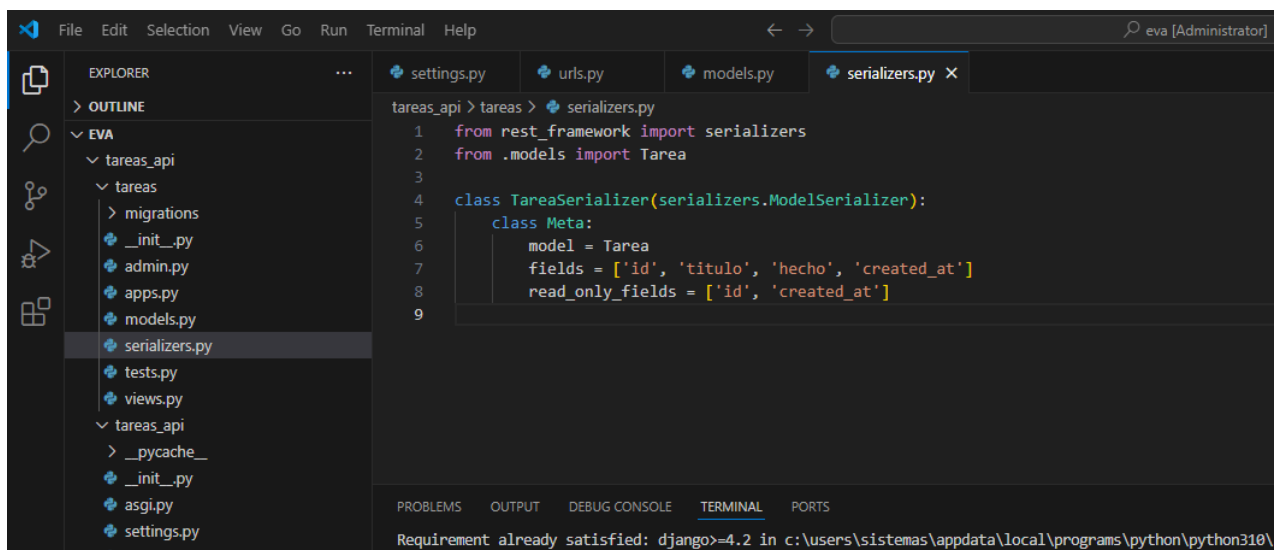
The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer shows a project structure with a folder named 'EVA' containing a subfolder 'tareas_api'. Inside 'tareas_api', there are files: 'tareas', 'tareas_api', '._pycache_', '._init_.py', 'asgi.py', 'settings.py', 'urls.py' (selected), 'wsgi.py', and 'manage.py'. The main editor area displays the content of 'urls.py' in the 'tareas_api' app. The code includes a comment 'Including another URLconf', a path for 'blog/' that includes 'blog.urls', and a list of URL patterns for 'admin/' and 'api/v1/'.

```
13 Including another URLconf
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, include
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('api/v1/', include('tareas.urls')), # versionado
23 ]
24
```



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer shows the same project structure as the previous screenshot, but now 'models.py' is selected in the 'tareas_api' folder. The main editor area displays the content of 'models.py'. It imports 'models' from 'django.db', defines a 'Tarea' model with fields 'titulo', 'hecho', and 'created_at', and a 'Meta' class with 'ordering'.

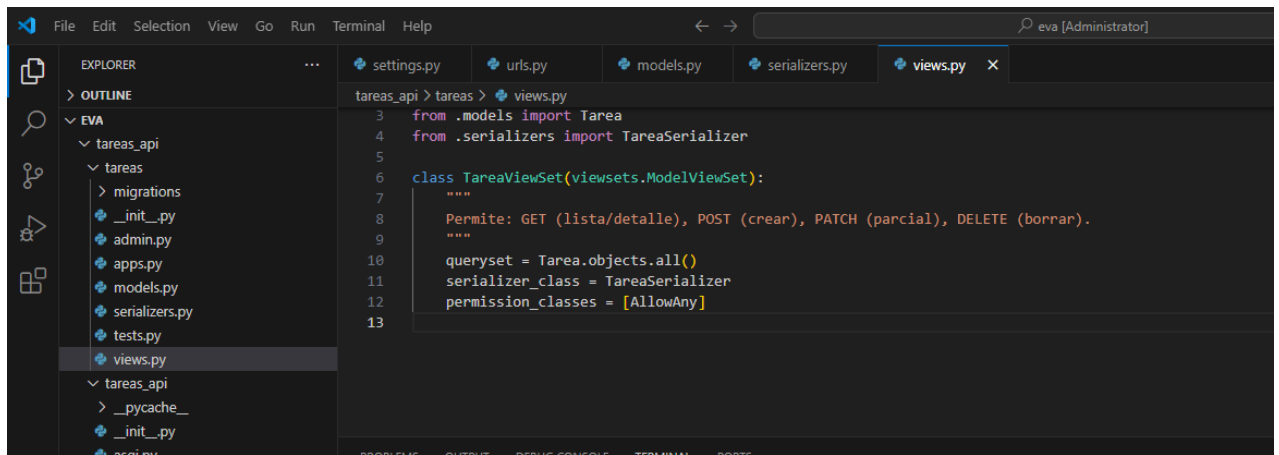
```
1 from django.db import models
2
3 class Tarea(models.Model):
4     titulo = models.CharField(max_length=255)
5     hecho = models.BooleanField(default=False)
6     created_at = models.DateTimeField(auto_now_add=True)
7
8     class Meta:
9         ordering = ['-created_at']
10
11     def __str__(self):
12         return f"{self.id} - {self.titulo}"
13
```



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer shows the same project structure, but now 'serializers.py' is selected in the 'tareas_api' folder. The main editor area displays the content of 'serializers.py'. It imports 'serializers' from 'rest_framework' and 'Tarea' from '.models', and defines a 'TareaSerializer' class that inherits from 'ModelSerializer'.

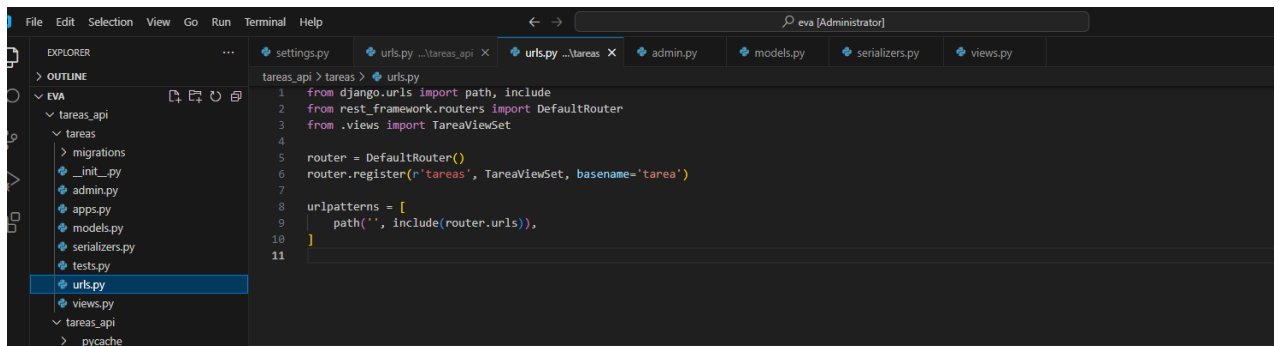
```
1 from rest_framework import serializers
2 from .models import Tarea
3
4 class TareaSerializer(serializers.ModelSerializer):
5     class Meta:
6         model = Tarea
7         fields = ['id', 'titulo', 'hecho', 'created_at']
8         read_only_fields = ['id', 'created_at']
9
```

At the bottom of the editor, the 'TERMINAL' tab is active, showing the output of a command: 'Requirement already satisfied: django=4.2 in c:\users\sistemas\appdata\local\programs\python\python310\'. The terminal also shows tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'PORTS'.



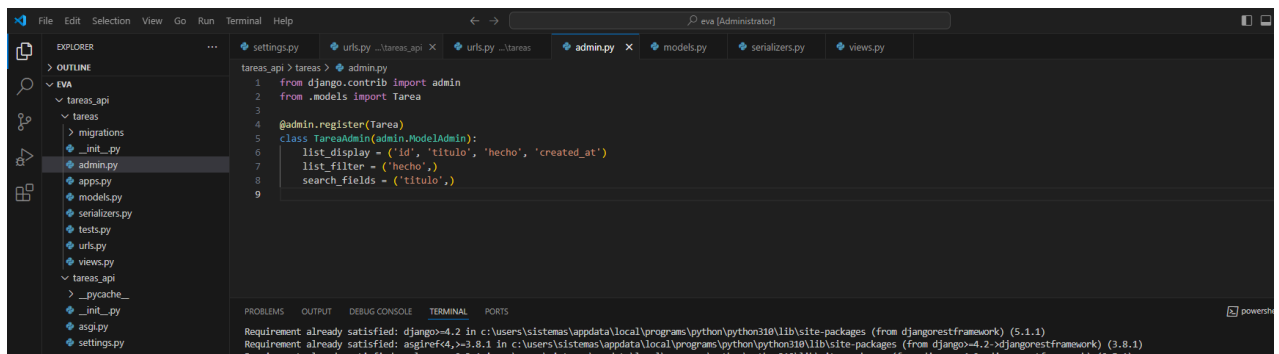
The VS Code editor shows the Explorer sidebar with the project structure. The file explorer is expanded to show the 'views.py' file under the 'tareas' directory. The main editor window displays the code for 'views.py'.

```
1 from .models import Tarea
2 from .serializers import TareaSerializer
3
4 class TareaViewSet(viewsets.ModelViewSet):
5     """
6     Permitir: GET (lista/detalle), POST (crear), PATCH (parcial), DELETE (borrar).
7     """
8     queryset = Tarea.objects.all()
9     serializer_class = TareaSerializer
10    permission_classes = [AllowAny]
```



The VS Code editor shows the Explorer sidebar with the project structure. The file explorer is expanded to show the 'urls.py' file under the 'tareas' directory. The main editor window displays the code for 'urls.py'.

```
1 from django.urls import path, include
2 from rest_framework.routers import DefaultRouter
3 from .views import TareaViewSet
4
5 router = DefaultRouter()
6 router.register(r'tareas', TareaViewSet, basename='tarea')
7
8 urlpatterns = [
9     path('', include(router.urls)),
10 ]
```



The VS Code editor shows the Explorer sidebar with the project structure. The file explorer is expanded to show the 'admin.py' file under the 'tareas' directory. The main editor window displays the code for 'admin.py'.

```
1 from django.contrib import admin
2 from .models import Tarea
3
4 @admin.register(Tarea)
5 class TareaAdmin(admin.ModelAdmin):
6     list_display = ('id', 'titulo', 'hecho', 'created_at')
7     list_filter = ('hecho',)
8     search_fields = ('titulo',)
```

At the bottom of the editor, the terminal shows the following output:

```
Requirement already satisfied: django==4.2 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from djangorestframework) (5.1.1)
Requirement already satisfied: asgiref<4.0, >=3.8.1 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from django==4.2->djangorestframework) (3.8.1)
Requirement already satisfied: sqlparse==0.4.4 in c:\users\sistemas\appdata\local\programs\python\python310\lib\site-packages (from django==4.2->djangorestframework) (0.4.4)
```

```

▼ tareas_api
  > __pycache__
  * __init__.py
  * asgi.py
  * settings.py
  * urls.py
  * wsgi.py
  db.sqlite3
  * manage.py

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Successfully installed django-rest-framework-3.16.1

[notice] A new release of pip is available: 24.2 -> 25.2
[notice] To update, run: python.exe django-admin startproject tareas_api
PS C:\Users\sistemas\Desktop\eva> cd .\tareas_api
PS C:\Users\sistemas\Desktop\eva\tareas_api> python manage.py startapp tareas
PS C:\Users\sistemas\Desktop\eva\tareas_api> python manage.py makemigrations
Migrations for 'tareas':
  tareas\migrations\0001_initial.py
    + Create model Tarea
PS C:\Users\sistemas\Desktop\eva\tareas_api> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, tareas
Running migrations:
  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
  Applying tareas.0001_initial... OK
PS C:\Users\sistemas\Desktop\eva\tareas_api>

```

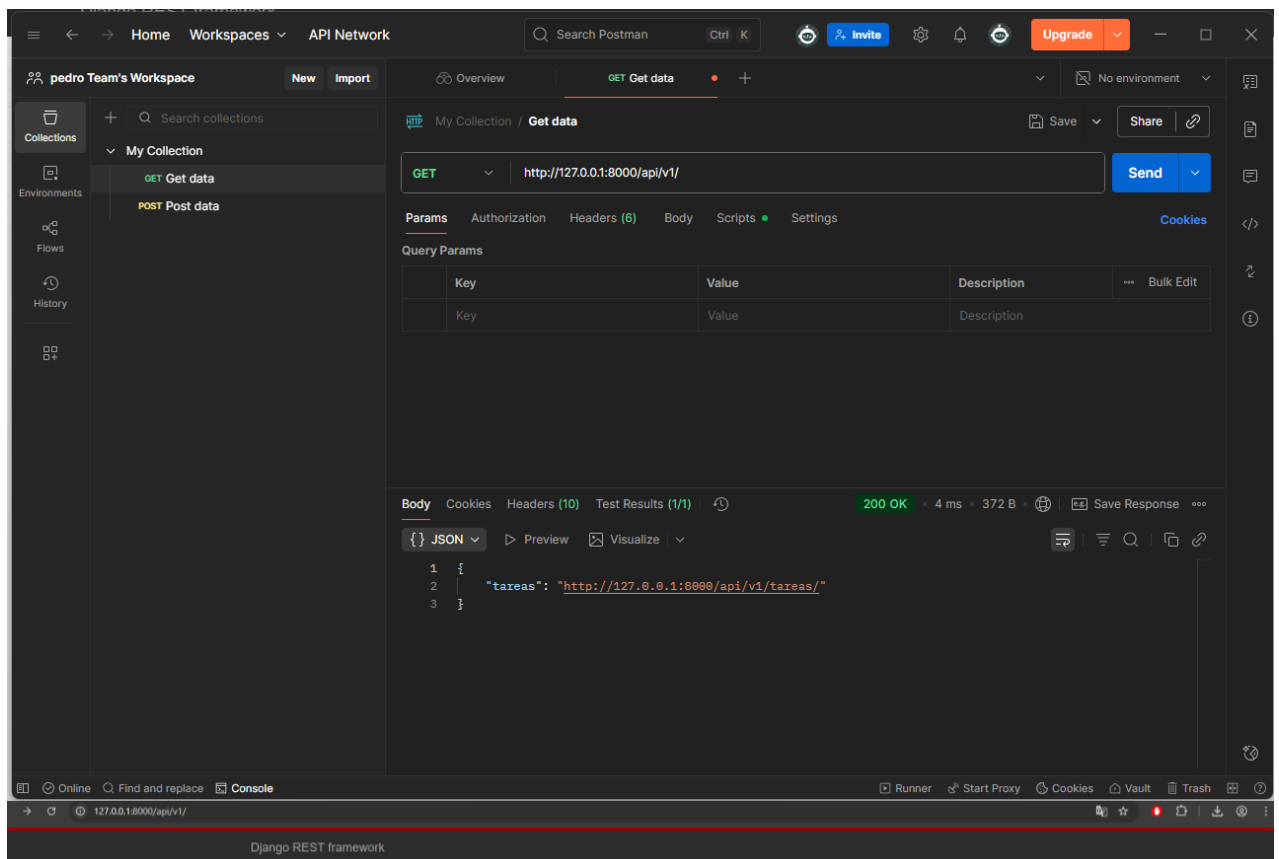
```

Applying tareas.0001_initial... OK
PS C:\Users\sistemas\Desktop\eva\tareas_api> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 13, 2025 - 21:01:39
Django version 5.1.1, using settings 'tareas_api.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

Not Found: /
[13/Oct/2025 21:01:41] "GET / HTTP/1.1" 404 2311
Not Found: /favicon.ico
[13/Oct/2025 21:01:41] "GET /favicon.ico HTTP/1.1" 404 2362
[13/Oct/2025 21:02:45] "GET /api/ HTTP/1.1" 200 5395

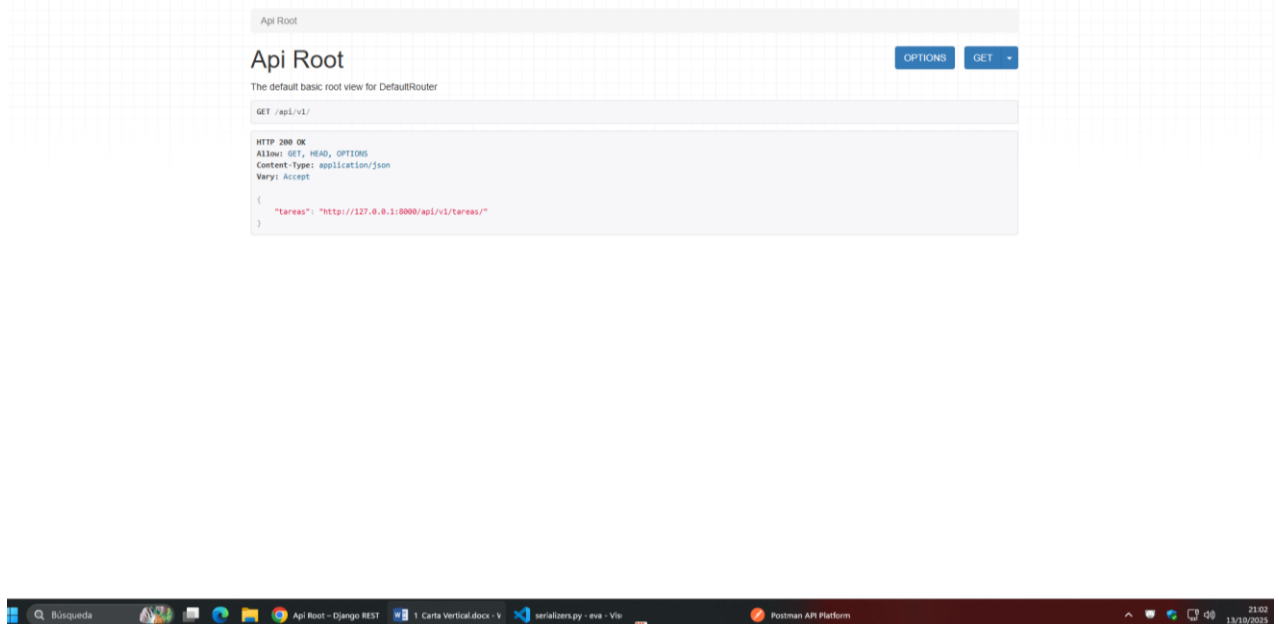
```



The screenshot shows the Postman API client interface. The top bar includes navigation links (Home, Workspaces, API Network), a search bar, and an 'Upgrade' button. The left sidebar shows the 'pedro Team's Workspace' with a 'My Collection' containing 'GET Get data' and 'POST Post data'. The main panel displays the 'GET Get data' request details, including the URL 'http://127.0.0.1:8000/api/v1/' and a 'Send' button. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Scripts', and 'Settings'. The 'Query Params' section is currently active, showing a table with columns 'Key', 'Value', and 'Description'. The 'Body' tab is also visible, showing a JSON response:

```
{ 1: { 2: { 3: { "tareas": "http://127.0.0.1:8000/api/v1/tareas/" } } }
```

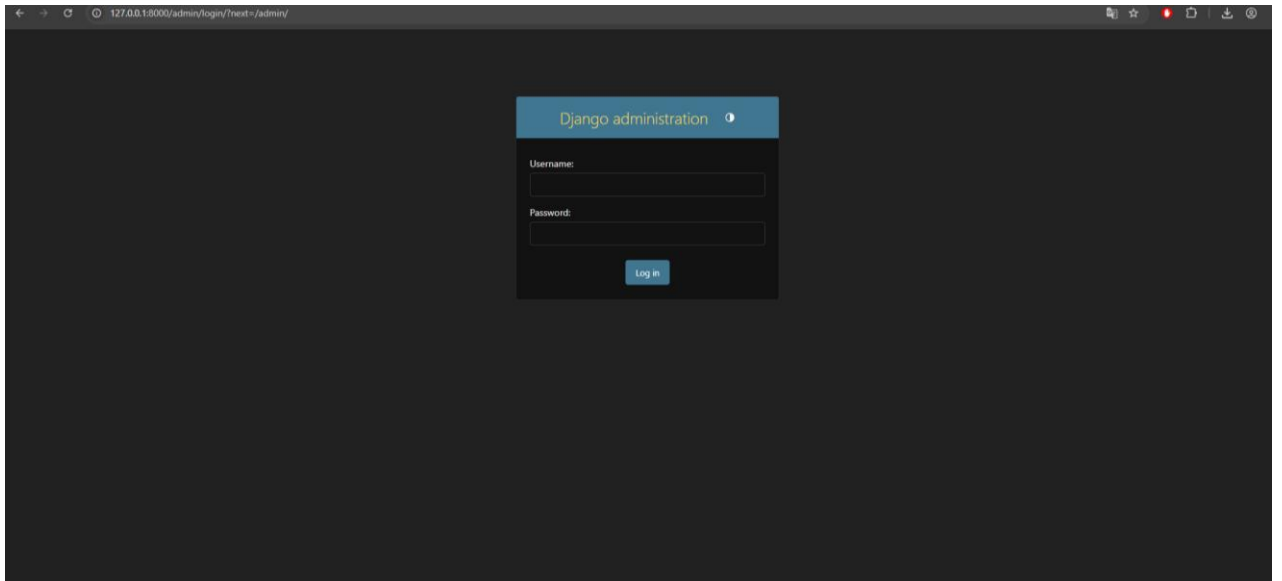
. The status bar at the bottom indicates a '200 OK' response with a 4 ms latency and 372 B of data.



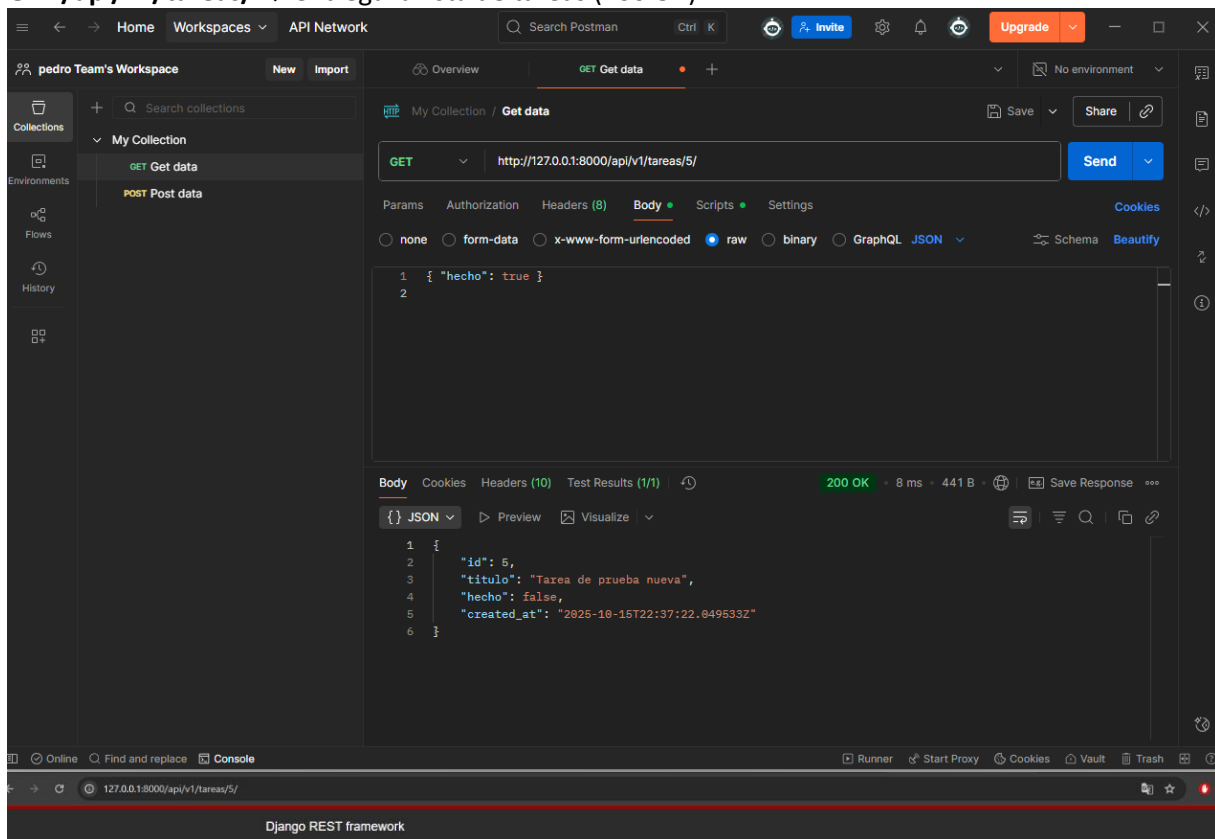
The screenshot shows the Django REST framework 'Api Root' view. The page title is 'Api Root' and it includes a description: 'The default basic root view for DefaultRouter'. There are two buttons, 'OPTIONS' and 'GET', in the top right corner. The main content area displays the response for the 'GET /api/v1/' endpoint, which is a 200 OK status with a 'Content-Type: application/json' header. The response body is a JSON object:

```
{  "tareas": "http://127.0.0.1:8000/api/v1/tareas/" }
```

. The bottom of the image shows a Windows taskbar with several open applications, including 'Busqueda', 'Api Root - Django REST', 'Carta Vertical.docx', 'serializers.py - vs - Vie', and 'Postman API Platform'.

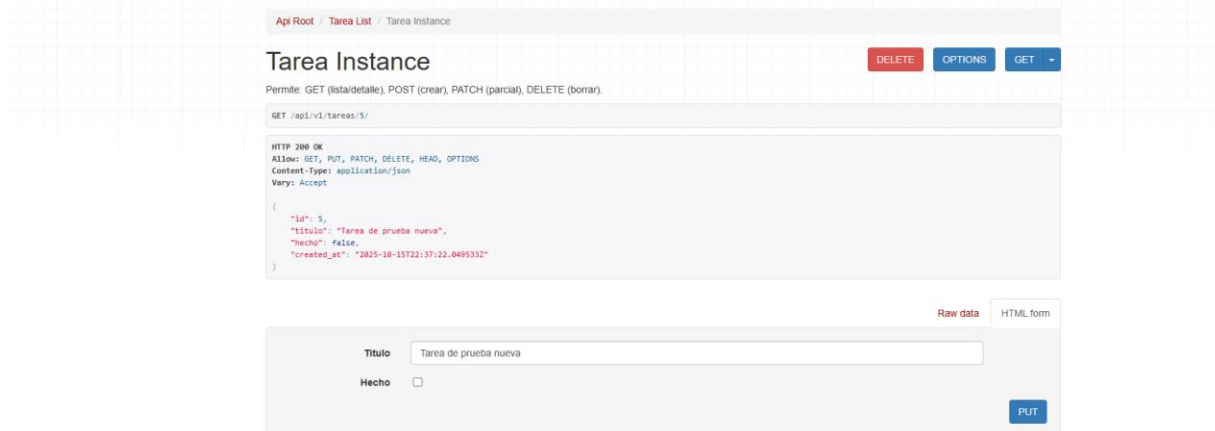


GET /api/v1/tareas/ → entrega la lista de tareas (200 OK).



The screenshot shows the Postman interface. The request is a GET to `http://127.0.0.1:8000/api/v1/tareas/5/`. The response is a 200 OK with a body containing the following JSON:

```
{
  "id": 5,
  "titulo": "Tarea de prueba nueva",
  "hecho": false,
  "created_at": "2025-10-15T22:37:22.049533Z"
}
```

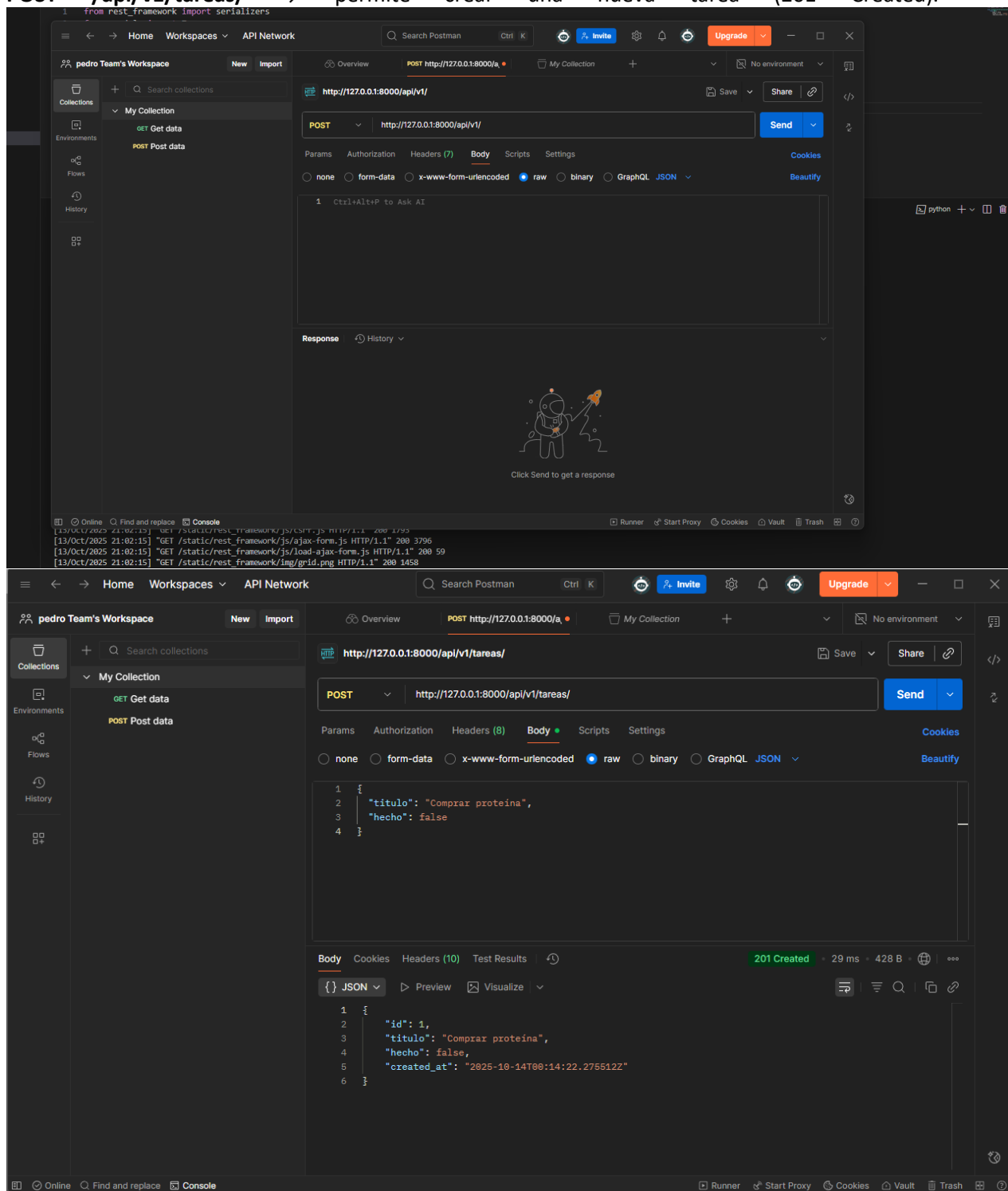


The screenshot shows the Django REST framework API browser. The page title is 'Tarea Instance'. The URL is `GET /api/v1/tareas/5/`. The response is an HTTP 200 OK with the following JSON:

```
{
  "id": 5,
  "titulo": "Tarea de prueba nueva",
  "hecho": false,
  "created_at": "2025-10-15T22:37:22.049533Z"
}
```

Below the JSON, there is a form to update the instance. The 'Titulo' field is 'Tarea de prueba nueva' and the 'Hecho' checkbox is unchecked. A 'PUT' button is visible.

POST `/api/v1/tareas/` → permite crear una nueva tarea (201 Created).



The image displays two screenshots of the Postman API client interface, demonstrating a REST API endpoint for creating tasks.

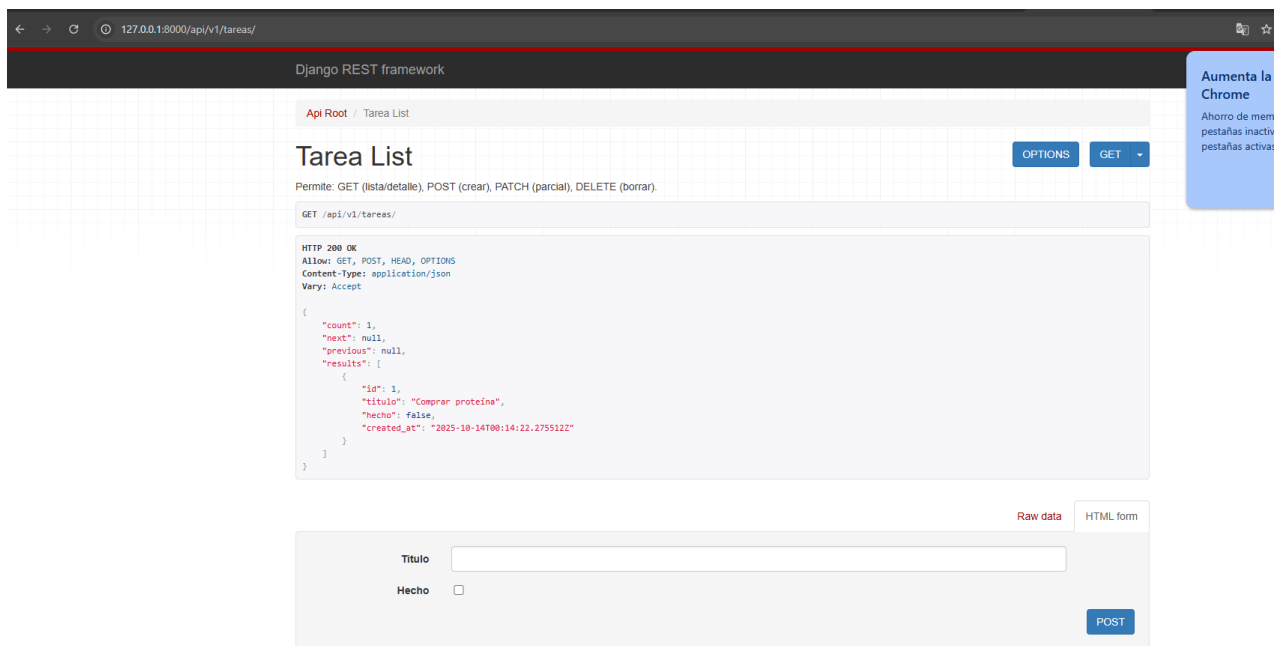
Top Screenshot: Shows a POST request to `http://127.0.0.1:8000/api/v1/`. The request body is empty. The response is a 201 Created status, indicated by a green message at the bottom: "201 Created".

Bottom Screenshot: Shows a POST request to `http://127.0.0.1:8000/api/v1/tareas/`. The request body is a JSON object:

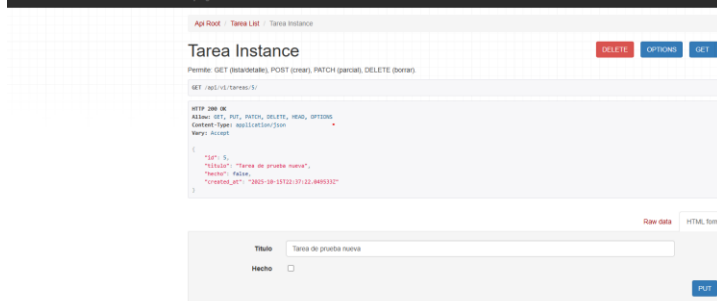
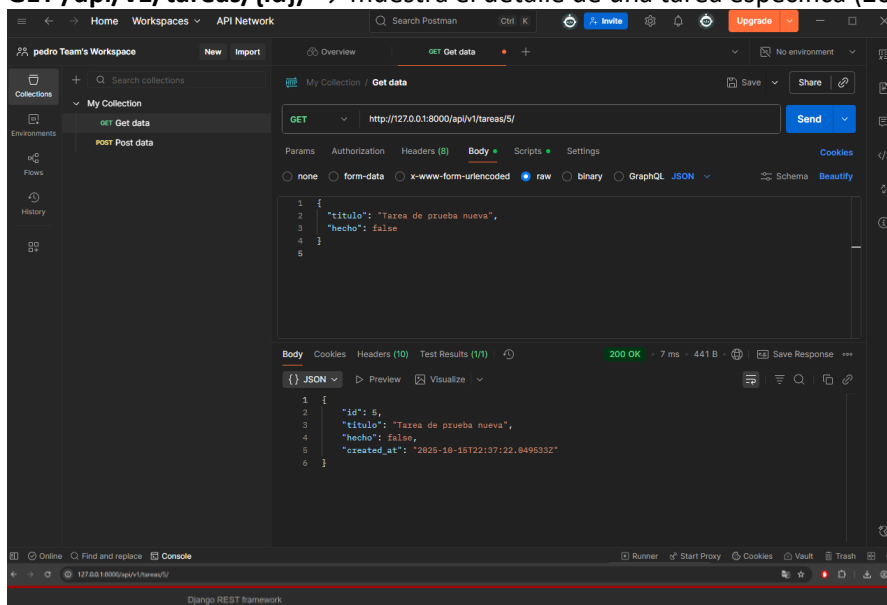
```
1 {
2   "titulo": "Comprar proteina",
3   "hecho": false
4 }
```

The response is a 201 Created status, indicated by a green message at the bottom: "201 Created". The response body is a JSON object:

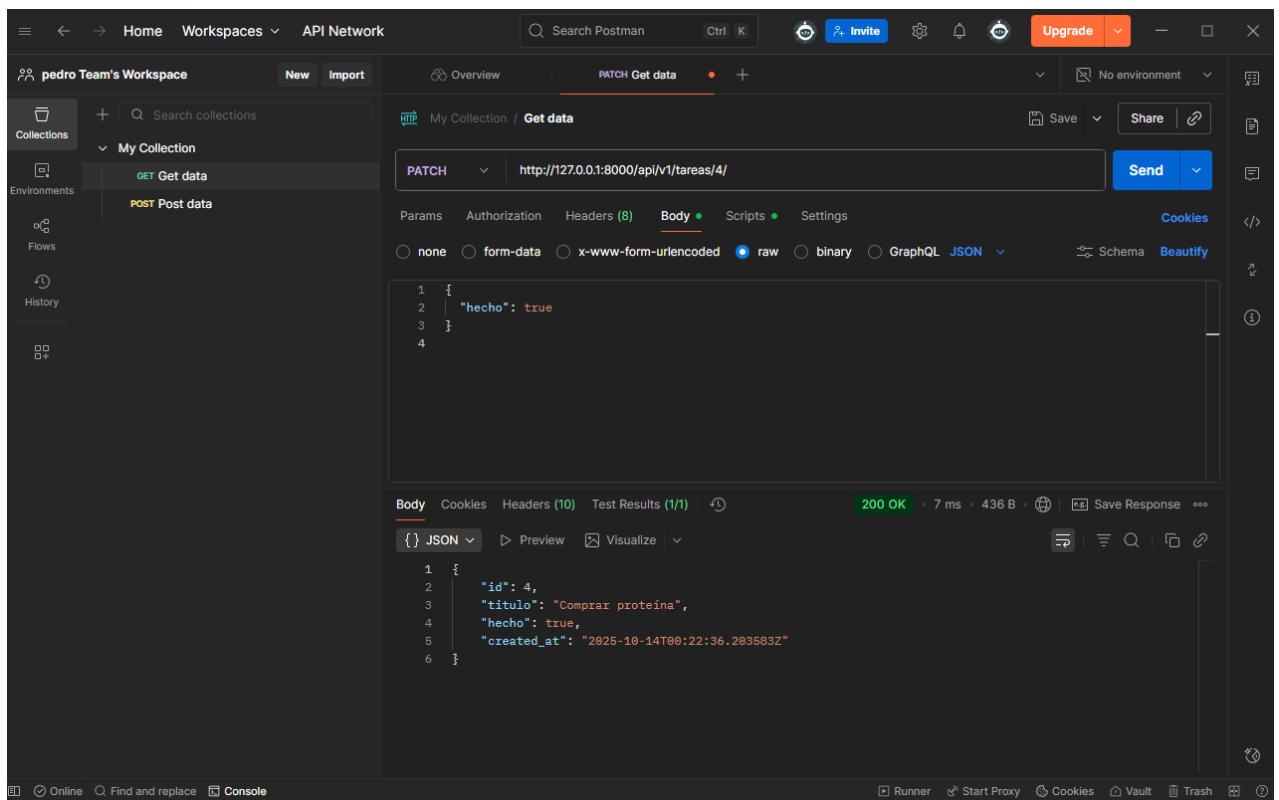
```
1 {
2   "id": 1,
3   "titulo": "Comprar proteina",
4   "hecho": false,
5   "created_at": "2025-10-14T08:14:22.275512Z"
6 }
```



GET /api/v1/tareas/{id}/ → muestra el detalle de una tarea específica (200 OK).



PATCH /api/v1/tareas/{id}/ → actualiza parcialmente el atributo hecho (200 OK).



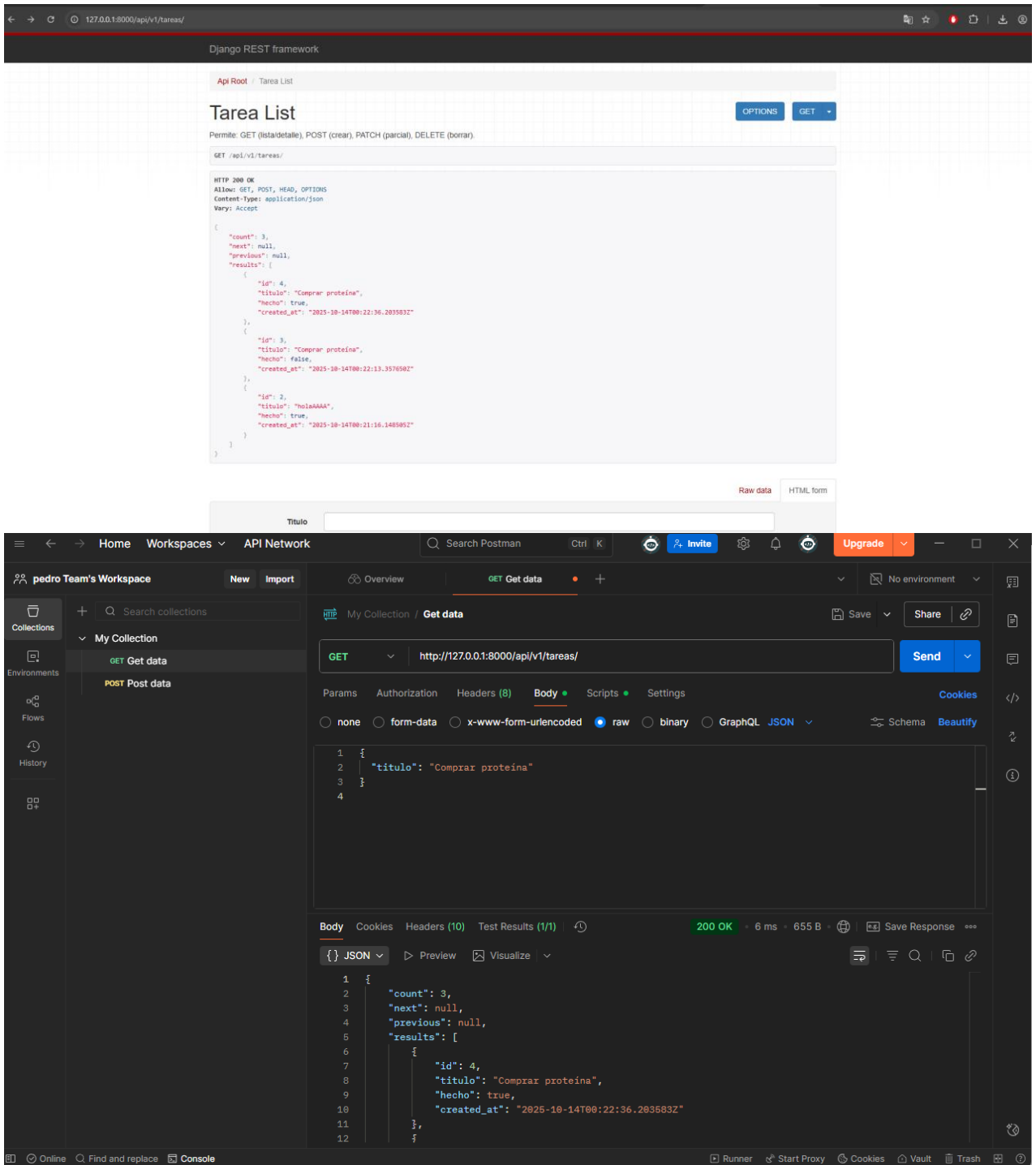
The image shows the Postman application interface. The top bar includes navigation links (Home, Workspaces, API Network), a search bar, and an 'Upgrade' button. The left sidebar shows the 'pedro Team's Workspace' with a 'My Collection' containing 'GET Get data' and 'POST Post data'. The main area displays a PATCH request to 'http://127.0.0.1:8000/api/v1/tareas/4'. The request body is a JSON object:

```
{  "hecho": true}
```

. The response is a 200 OK status with a 7 ms response time and 436 B of data. The response body is a JSON object:

```
{  "id": 4,  "titulo": "Comprar proteina",  "hecho": true,  "created_at": "2025-10-14T00:22:36.203583Z"}
```

. The bottom status bar shows 'Online', 'Find and replace', 'Console', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a help icon.



The image shows two screenshots related to a Django REST framework API.

Top Screenshot (Web Browser): Displays the Django REST framework interface for the `Tarea List` endpoint. The URL is `127.0.0.1:8000/api/v1/tareas/`. The interface shows the allowed methods (GET, POST, PATCH, DELETE) and the response body for a GET request. The response is a JSON object:

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "count": 3,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 4,
      "titulo": "Comprar proteina",
      "hecho": true,
      "created_at": "2025-10-14T00:22:36.203583Z"
    },
    {
      "id": 3,
      "titulo": "Comprar proteina",
      "hecho": false,
      "created_at": "2025-10-14T00:22:13.357690Z"
    },
    {
      "id": 2,
      "titulo": "HolaAAAA",
      "hecho": true,
      "created_at": "2025-10-14T00:21:16.148595Z"
    }
  ]
}
```

Bottom Screenshot (Postman): Shows the Postman client interface. The request is a GET request to `http://127.0.0.1:8000/api/v1/tareas/`. The response is a 200 OK status with a response time of 6 ms and a body size of 655 B. The response body is a JSON object:

```
{
  "count": 3,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 4,
      "titulo": "Comprar proteina",
      "hecho": true,
      "created_at": "2025-10-14T00:22:36.203583Z"
    },
    {
      "id": 3,
      "titulo": "Comprar proteina",
      "hecho": false,
      "created_at": "2025-10-14T00:22:13.357690Z"
    },
    {
      "id": 2,
      "titulo": "HolaAAAA",
      "hecho": true,
      "created_at": "2025-10-14T00:21:16.148595Z"
    }
  ]
}
```

DELETE `/api/v1/tareas/{id}/` → elimina la tarea del sistema (204 No Content).

