

## **Equipo 8 - ciclo 4**

### **Integrantes:**

Raúl Balaguera

María González

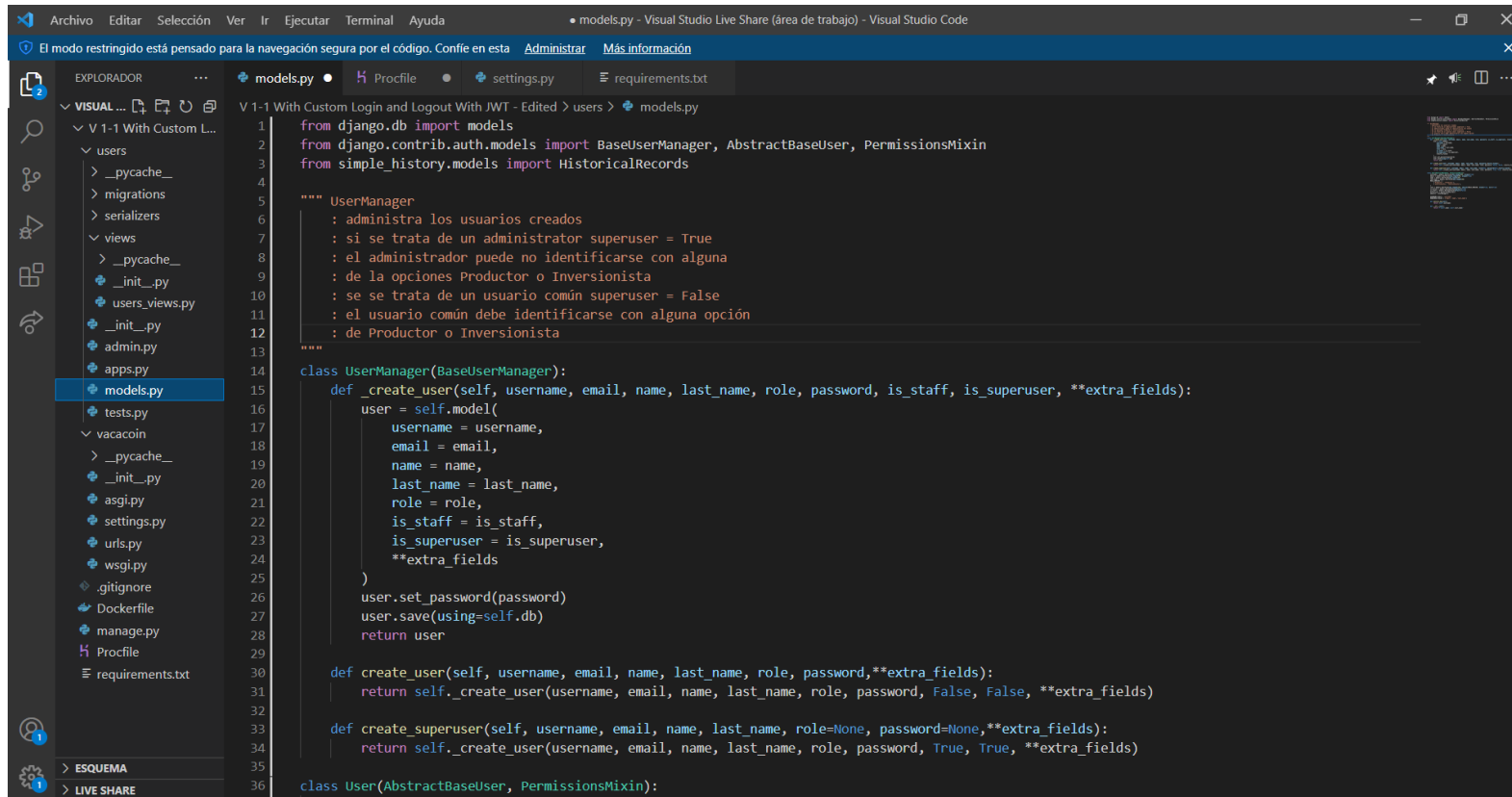
Santiago Jaramillo

Santiago Sanabria

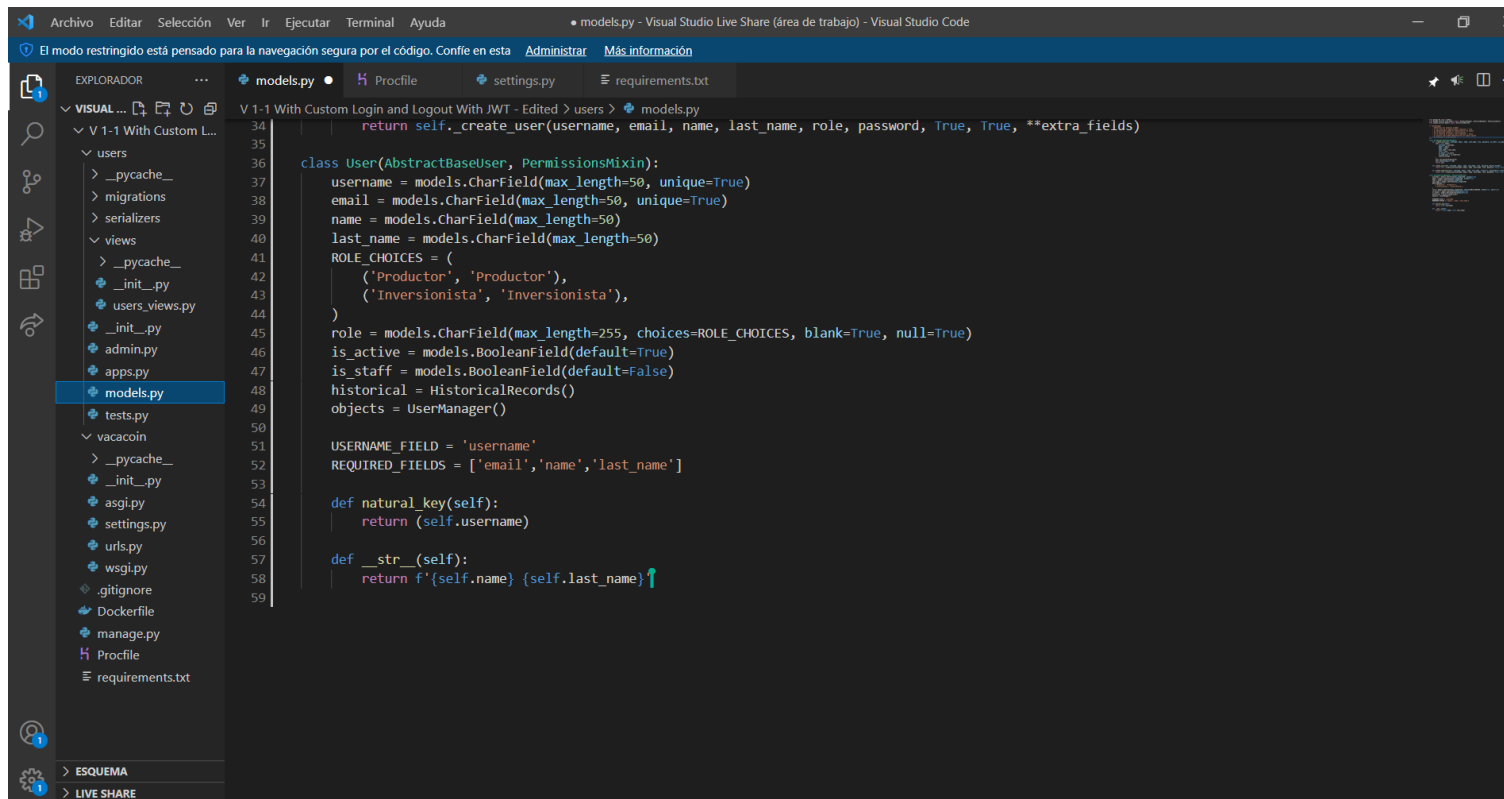
Mauricio Murcia

Daniel Pinto

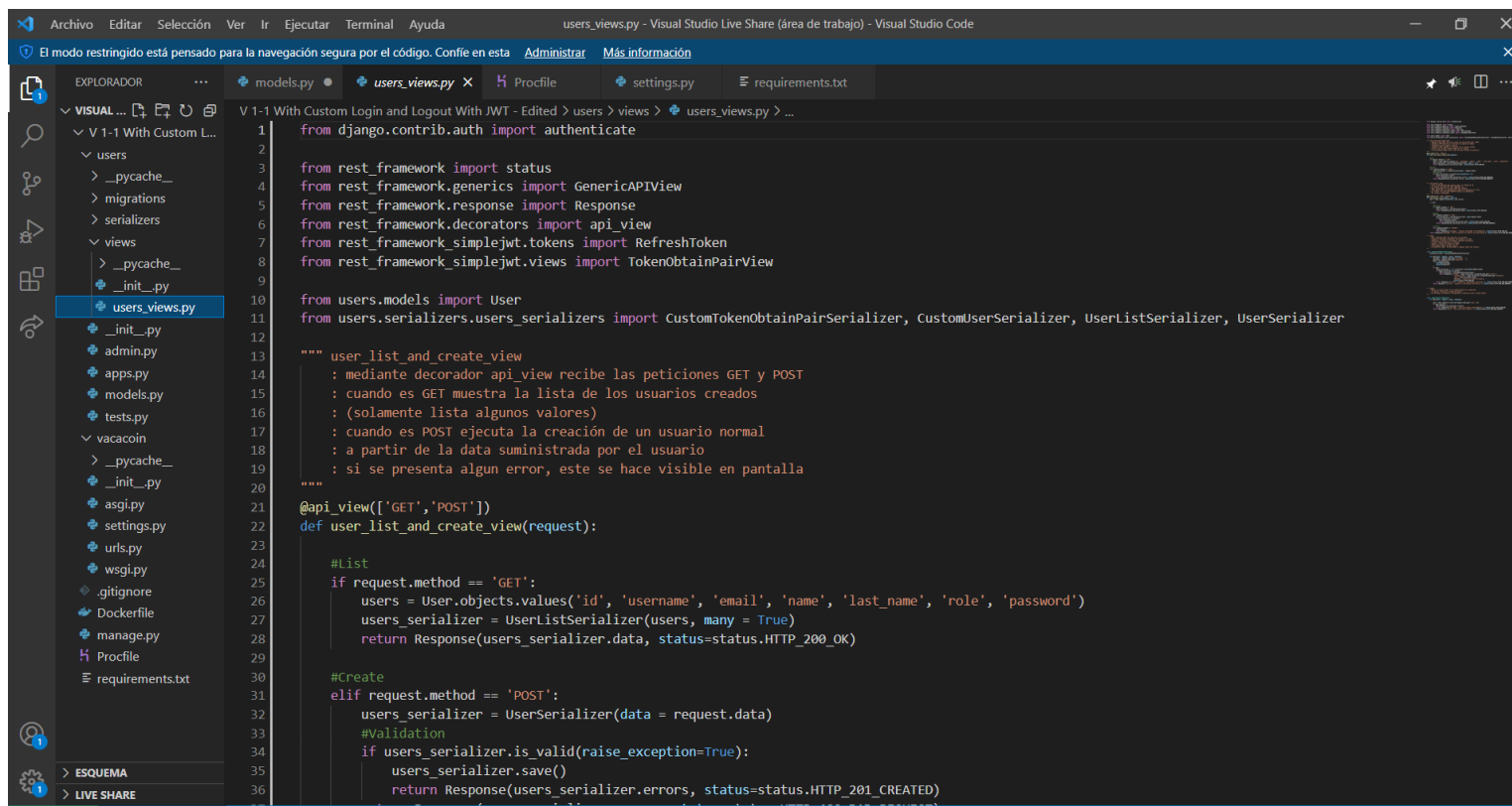
# MODELOS



```
1 from django.db import models
2 from django.contrib.auth.models import BaseUserManager, AbstractBaseUser, PermissionsMixin
3 from simple_history.models import HistoricalRecords
4
5 """
6 : administra los usuarios creados
7 : si se trata de un administrador superuser = True
8 : el administrador puede no identificarse con alguna
9 : de la opciones Productor o Inversionista
10 : se se trata de un usuario común superuser = False
11 : el usuario común debe identificarse con alguna opción
12 : de Productor o Inversionista
13 """
14
15 class UserManager(BaseUserManager):
16     def _create_user(self, username, email, name, last_name, role, password, is_staff, is_superuser, **extra_fields):
17         user = self.model(
18             username = username,
19             email = email,
20             name = name,
21             last_name = last_name,
22             role = role,
23             is_staff = is_staff,
24             is_superuser = is_superuser,
25             **extra_fields
26         )
27         user.set_password(password)
28         user.save(using=self.db)
29         return user
30
31     def create_user(self, username, email, name, last_name, role, password, **extra_fields):
32         return self._create_user(username, email, name, last_name, role, password, False, False, **extra_fields)
33
34     def create_superuser(self, username, email, name, last_name, role=None, password=None, **extra_fields):
35         return self._create_user(username, email, name, last_name, role, password, True, True, **extra_fields)
36
37 class User(AbstractBaseUser, PermissionsMixin):
```



# VISTAS



The image shows a Visual Studio Code editor window with the file `users_views.py` open. The editor is displaying Python code for a Django REST Framework view. The left sidebar shows the Explorer view with the project structure, including `users_views.py` selected. The top bar indicates the file is `users_views.py - Visual Studio Live Share (área de trabajo) - Visual Studio Code`. A status bar at the top indicates `El modo restringido está pensado para la navegación segura por el código. Confíe en esta Administración Más información`.

```
1 from django.contrib.auth import authenticate
2
3 from rest_framework import status
4 from rest_framework.generics import GenericAPIView
5 from rest_framework.response import Response
6 from rest_framework.decorators import api_view
7 from rest_framework_simplejwt.tokens import RefreshToken
8 from rest_framework_simplejwt.views import TokenObtainPairView
9
10 from users.models import User
11 from users.serializers.users_serializers import CustomTokenObtainPairSerializer, CustomUserSerializer, UserListSerializer, UserSerializer
12
13 """ user_list_and_create view
14 : mediante decorador api_view recibe las peticiones GET y POST
15 : cuando es GET muestra la lista de los usuarios creados
16 : (solamente lista algunos valores)
17 : cuando es POST ejecuta la creación de un usuario normal
18 : a partir de la data suministrada por el usuario
19 : si se presenta algun error, este se hace visible en pantalla
20 """
21 @api_view(['GET', 'POST'])
22 def user_list_and_create_view(request):
23
24     #List
25     if request.method == 'GET':
26         users = User.objects.values('id', 'username', 'email', 'name', 'last_name', 'role', 'password')
27         users_serializer = UserListSerializer(users, many = True)
28         return Response(users_serializer.data, status=status.HTTP_200_OK)
29
30     #Create
31     elif request.method == 'POST':
32         users_serializer = UserSerializer(data = request.data)
33         #validation
34         if users_serializer.is_valid(raise_exception=True):
35             users_serializer.save()
36             return Response(users_serializer.errors, status=status.HTTP_201_CREATED)
```

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda users\_views.py - Visual Studio Live Share (área de trabajo) - Visual Studio Code

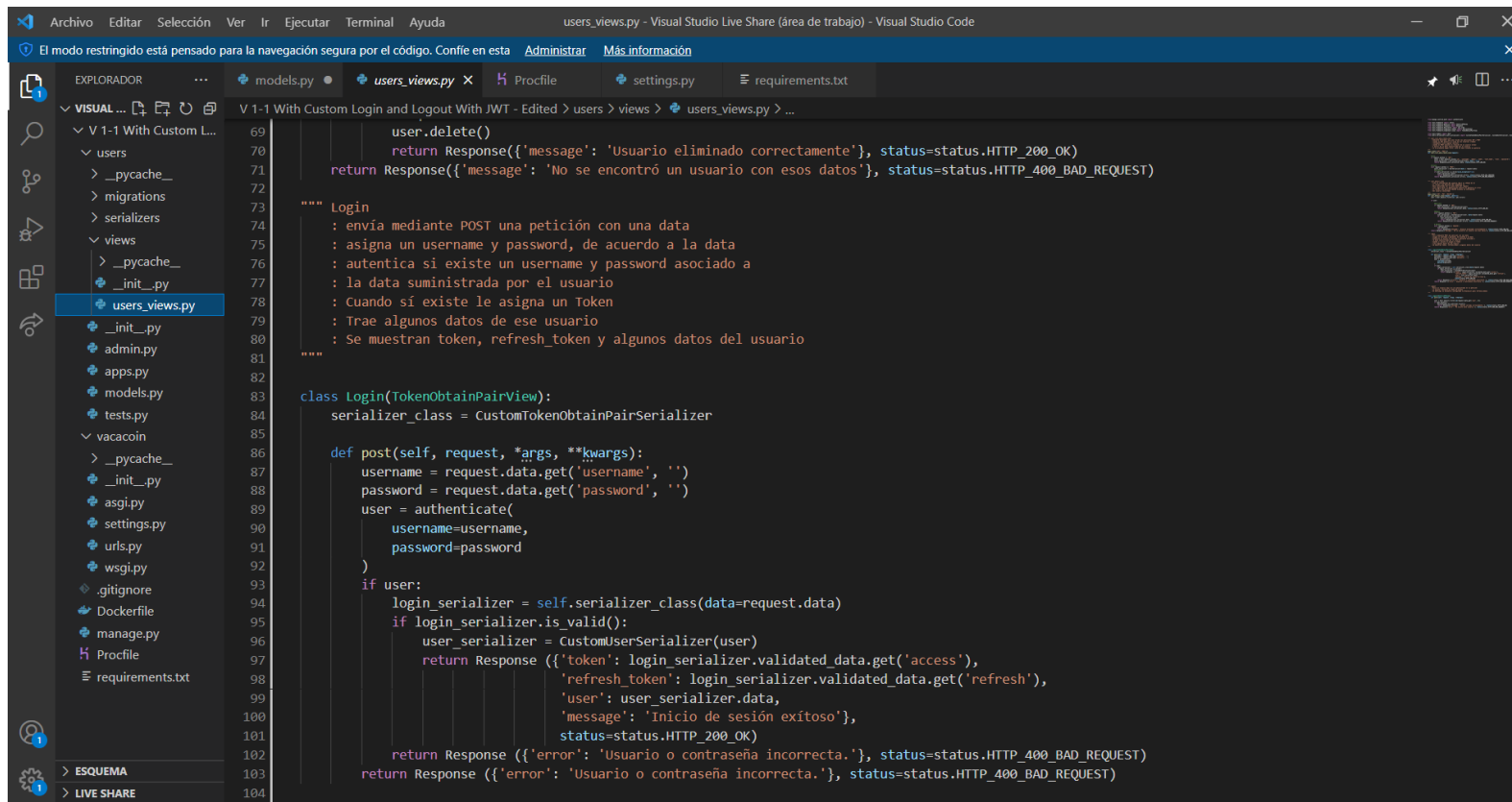
El modo restringido está pensado para la navegación segura por el código. Confié en esta Administración Más información

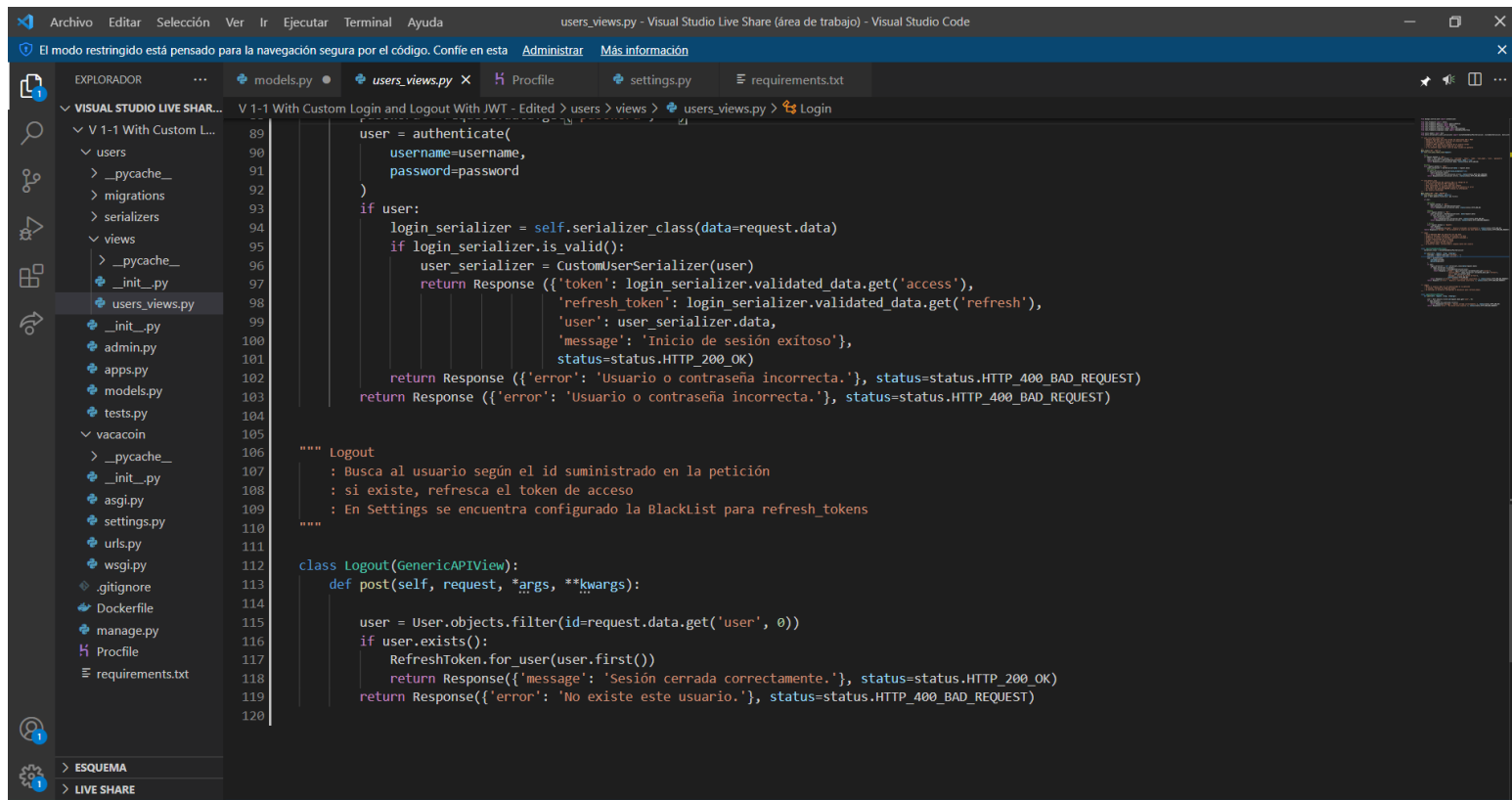
EXPLORADOR ... models.py users\_views.py Profile settings.py requirements.txt

VISUAL ... V 1-1 With Custom Login and Logout With JWT - Edited > users > views > users\_views.py > ...

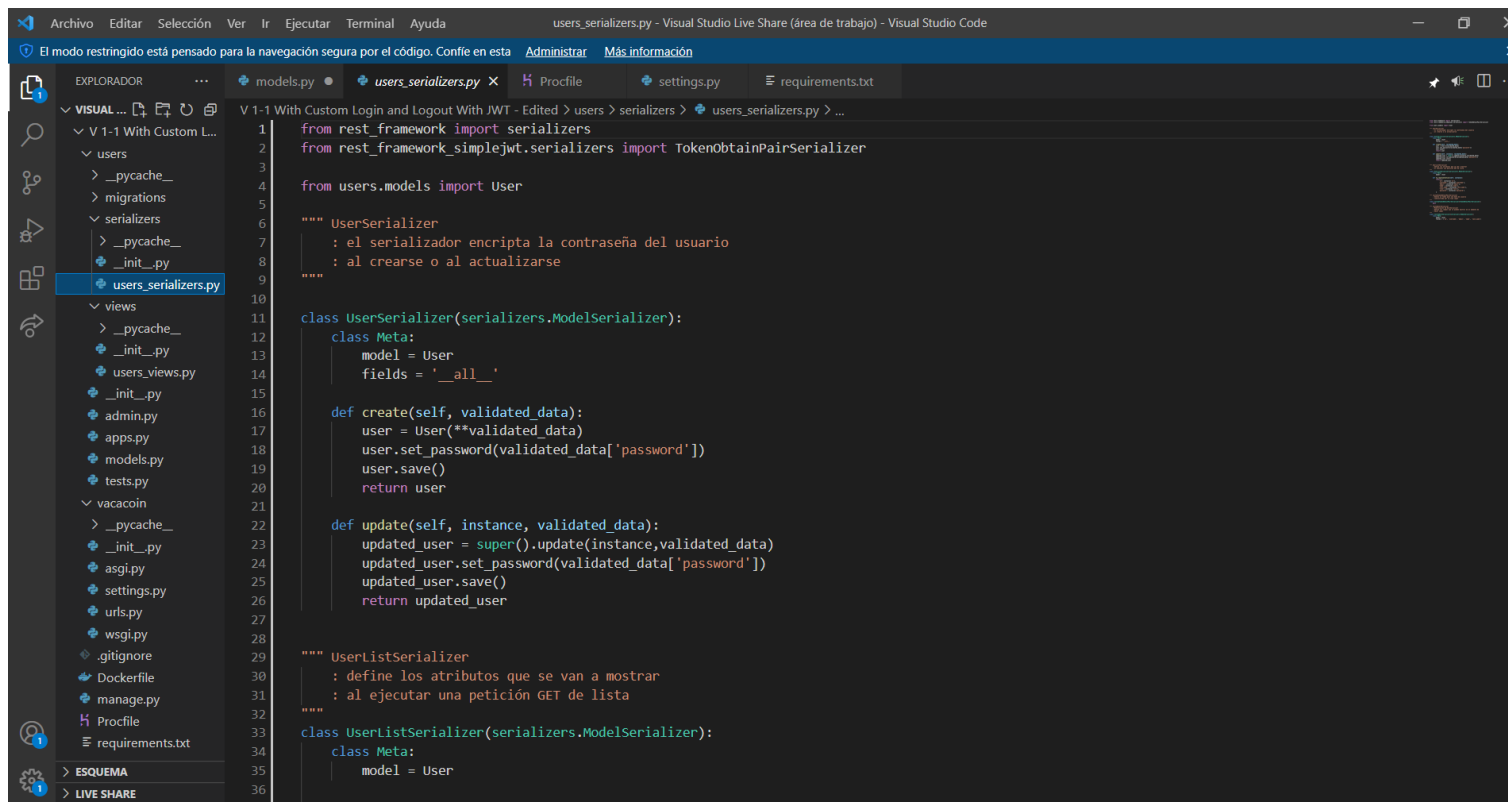
users\_views.py

```
36         return Response(users_serializer.errors, status=status.HTTP_201_CREATED)
37         return Response(users_serializer.errors, status=status.HTTP_400_BAD_REQUEST)
38
39
40     """ user_detail_view
41     : Trae la información del usuario según su código de id
42     : Al recibir una petición PUT reemplaza la
43     : data consultada con la nueva data del request
44     : sino logra hacer la modificación muestra en pantalla el error
45     : Al recibir una petición DELETE elimina la información
46     : del usuario consultado
47
48     """
49     @api_view(['GET', 'PUT', 'DELETE'])
50     def user_detail_view(request, pk=None):
51         user = User.objects.filter(id = pk).first()
52
53         if user:
54
55             #Retrieve
56             if request.method == 'GET':
57                 user_serializer = UserSerializer(user)
58                 return Response(user_serializer.data, status=status.HTTP_200_OK)
59
60             #Update
61             elif request.method == 'PUT':
62                 user_serializer = UserSerializer(user, data=request.data)
63                 if user_serializer.is_valid():
64                     user_serializer.save()
65                     return Response(user_serializer.data, status=status.HTTP_200_OK)
66                 return Response(user_serializer.errors, status=status.HTTP_400_BAD_REQUEST)
67
68             # Delete
69             elif request.method == 'DELETE':
70                 user.delete()
71                 return Response({'message': 'Usuario eliminado correctamente'}, status=status.HTTP_200_OK)
72         return Response({'message': 'No se encontró un usuario con esos datos'}, status=status.HTTP_400_BAD_REQUEST)
```



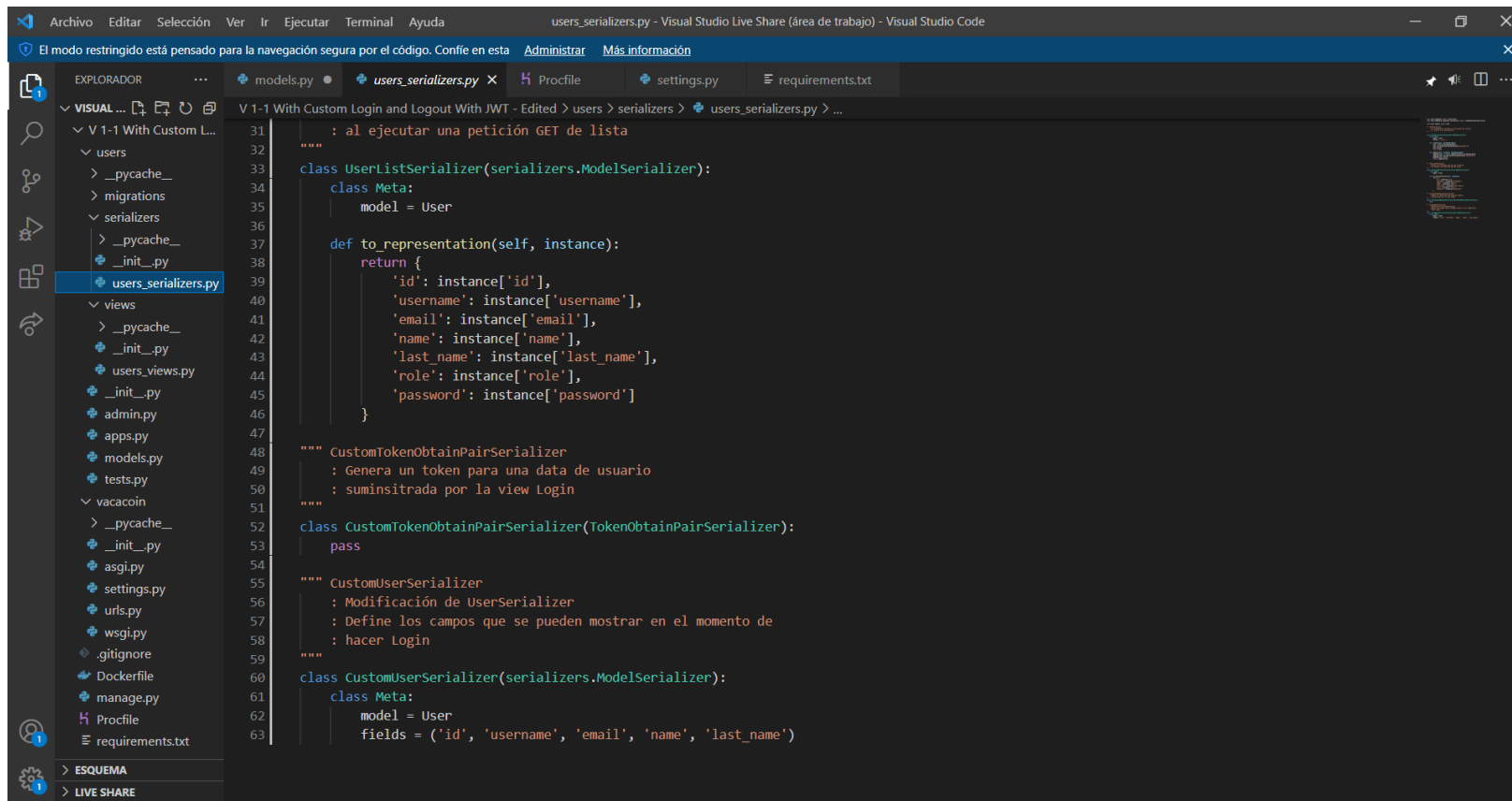


# SERIALIZERS



```
1 from rest_framework import serializers
2 from rest_framework_simplejwt.serializers import TokenObtainPairSerializer
3
4 from users.models import User
5
6 """ UserSerializer
7 : el serializador encripta la contraseña del usuario
8 : al crearse o al actualizarse
9 """
10
11 class UserSerializer(serializers.ModelSerializer):
12     class Meta:
13         model = User
14         fields = '__all__'
15
16     def create(self, validated_data):
17         user = User(**validated_data)
18         user.set_password(validated_data['password'])
19         user.save()
20         return user
21
22     def update(self, instance, validated_data):
23         updated_user = super().update(instance, validated_data)
24         updated_user.set_password(validated_data['password'])
25         updated_user.save()
26         return updated_user
27
28
29 """ UserListSerializer
30 : define los atributos que se van a mostrar
31 : al ejecutar una petición GET de lista
32 """
33
34 class UserListSerializer(serializers.ModelSerializer):
35     class Meta:
36         model = User
```





# URLS

```
1 """vacacoin 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 from django.contrib import admin
17 from django.urls import path
18 from rest_framework_simplejwt.views import TokenObtainPairView, TokenRefreshView
19 from users.views.users_views import Login, Logout, user_list_and_create_view, user_detail_view
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23
24     #Users
25     path('users/', user_list_and_create_view),
26     path('users/<int:pk>', user_detail_view),
27     path('login/', Login.as_view()),
28     path('logout/', Logout.as_view()),
29     path('api/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
30     path('api/token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
31 ]
32
```

# POSTMAN PROBANDO LOS ENDPOINTS

The screenshot displays the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore', along with a search bar and buttons for 'Sign In' and 'Create Account'. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface shows a GET request to 'http://127.0.0.1:8000/users/'. The 'Params' tab is active, showing a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'. Below this, the 'Body' tab is selected, displaying a JSON response in 'Pretty' format. The response status is '200 OK' with a time of '882 ms' and a size of '487 B'. The JSON data represents a user object with fields like 'id', 'username', 'email', 'name', 'last\_name', 'role', and 'password'.

GET http://127.0.0.1:8000/users/

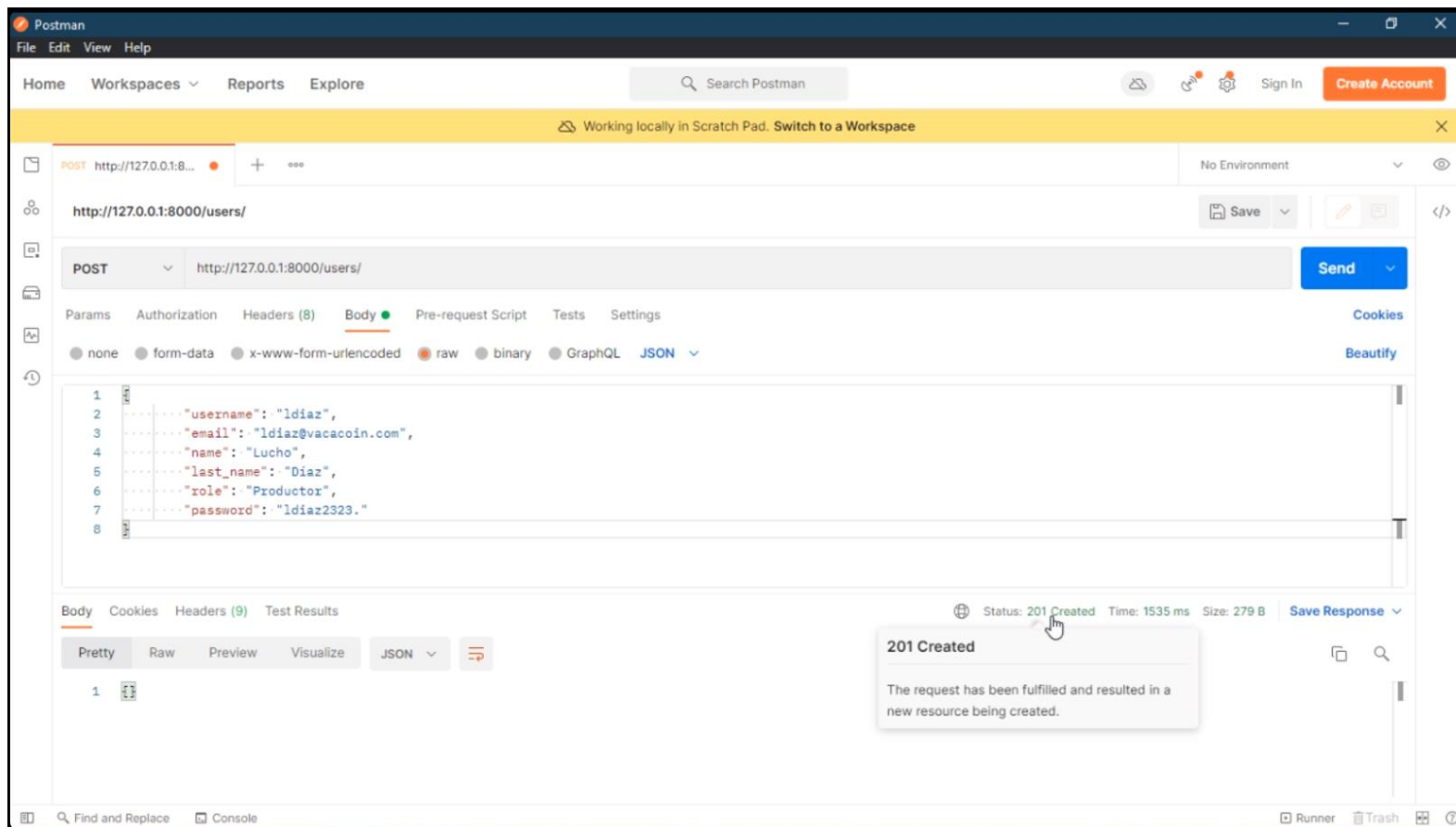
Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

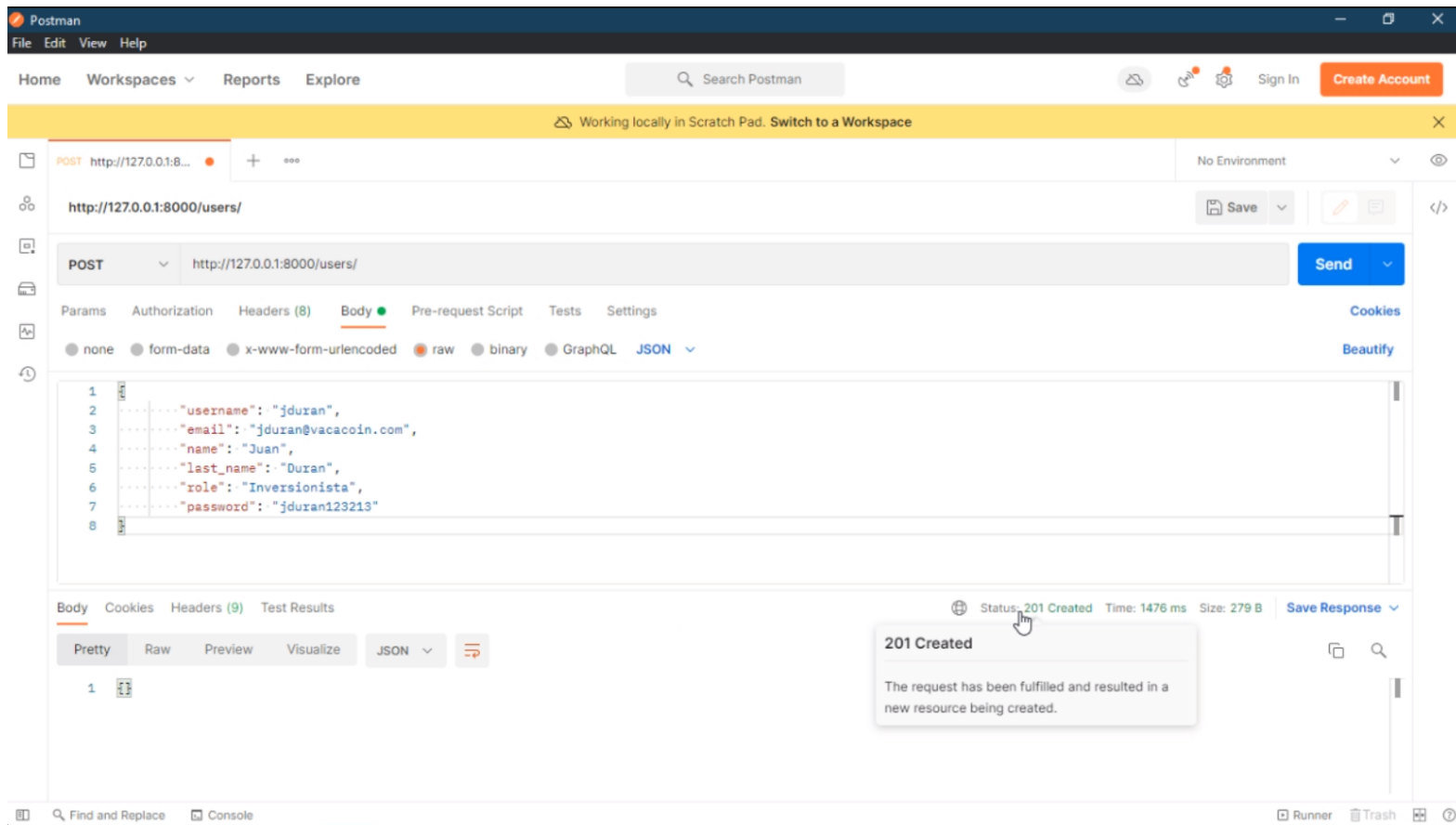
KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (9) Test Results

Status: 200 OK Time: 882 ms Size: 487 B Save Response

```
1 {
2   "id": 1,
3   "username": "admin",
4   "email": "admin@vacacoin.com",
5   "name": "administrator",
6   "last_name": "pro",
7   "role": null,
8   "password": "pbkdf2_sha256$260000$QxIoURtPtBkaWetXkr3Cx$YdBephF32bikD1#Opnykq3BU9ZRuhrPp318SCGDAYM="
9 }
10
11
```





Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

GET http://127.0.0.1:8000/users/ No Environment

Save

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (9) Test Results

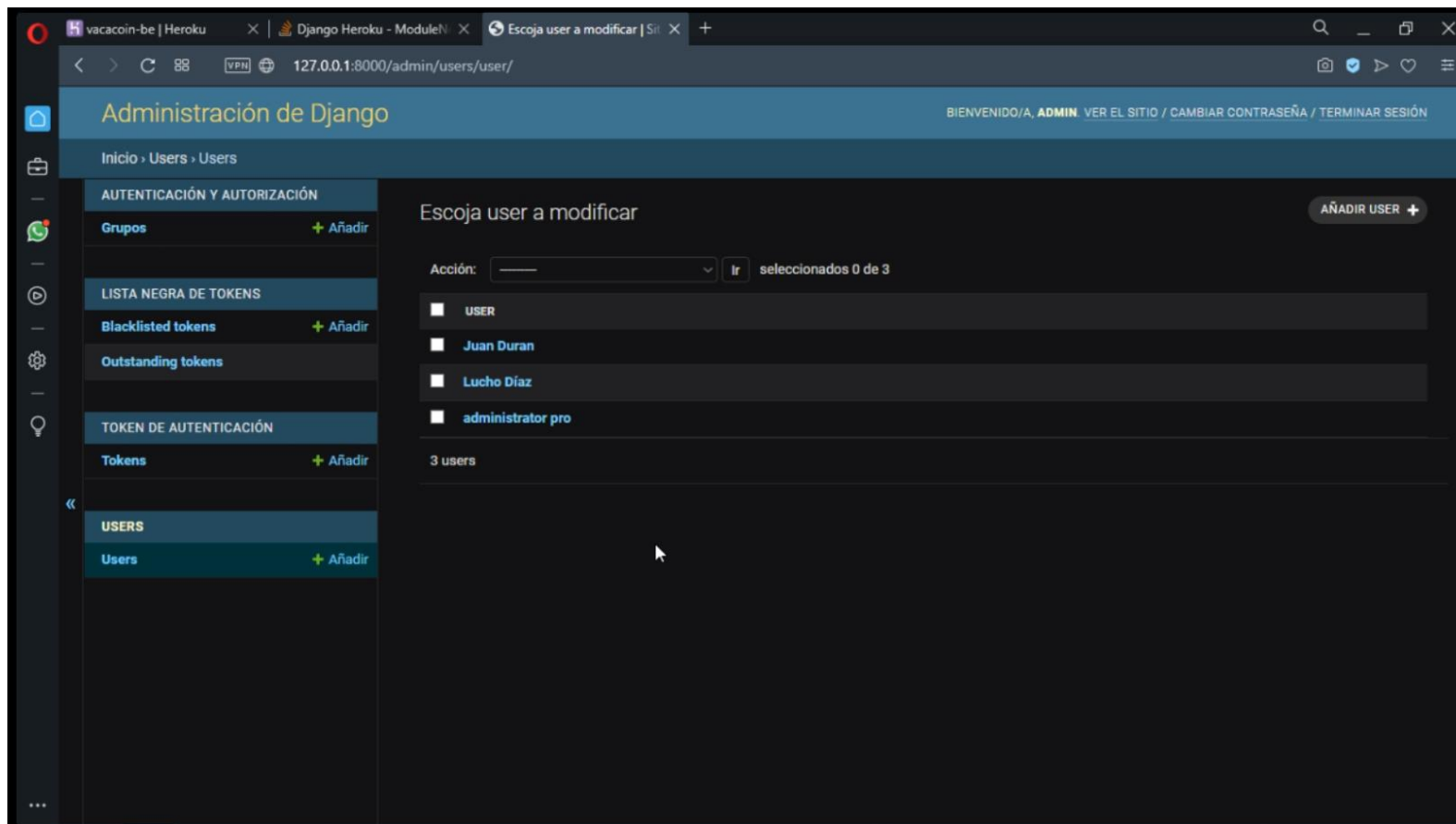
Status: 200 OK Time: 798 ms Size: 918 B Save Response

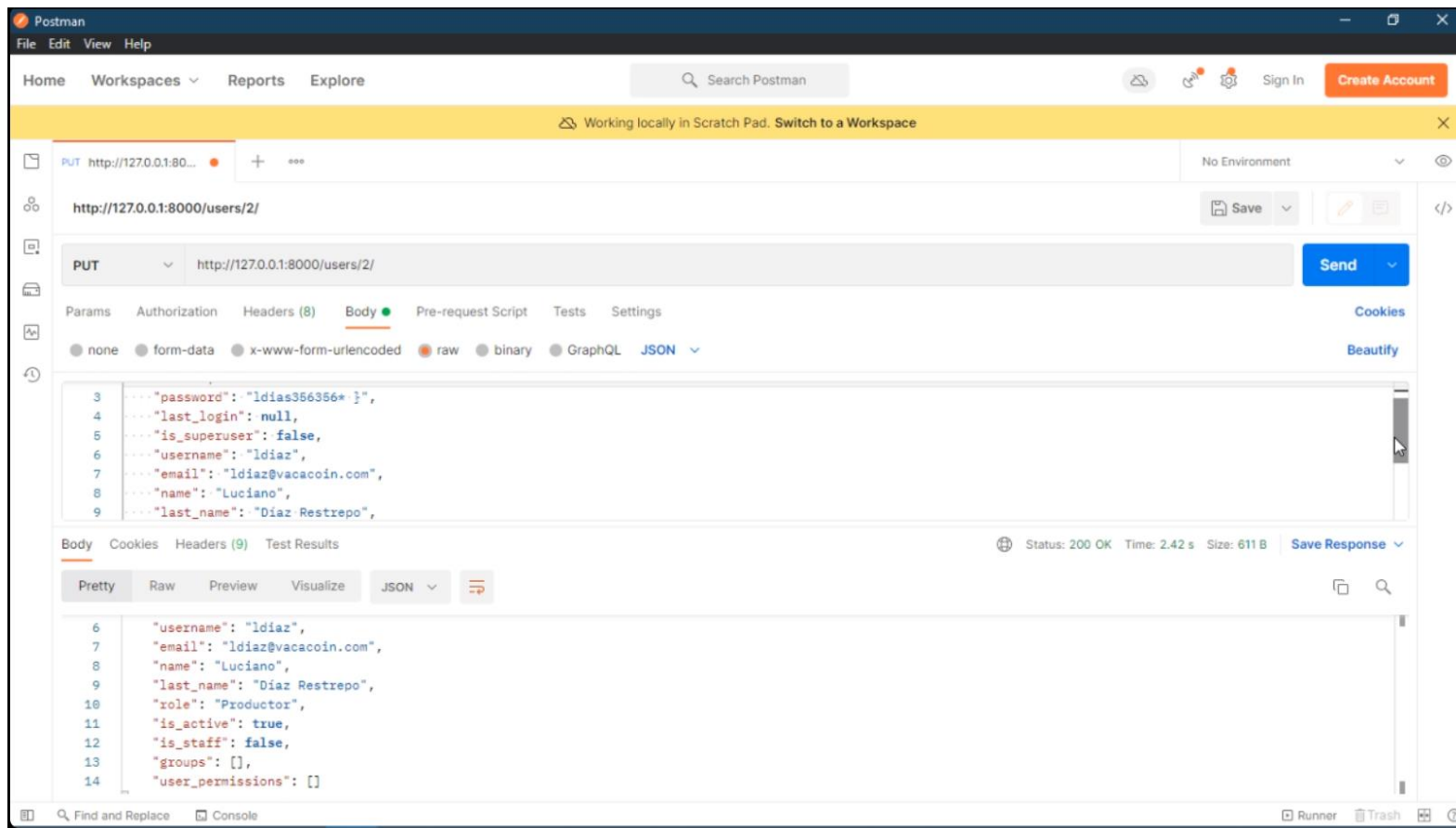
Pretty Raw Preview Visualize JSON

```
10  {
11    {
12      "id": 2,
13      "username": "ldiaz",
14      "email": "ldiaz@vacacoin.com",
15      "name": "Lucho",
16      "last_name": "Diaz",
17      "role": "Productor",
18      "password": "pbkdf2_sha256$260000$uVp44IN892LNs6h2t3xCch$JytcCEqAX7sYnKq07Ht6g+qdzkoICzvMftBz6W5vzP0="
19    },
20    {
21      "id": 3,
22      "username": "jduzan",
23      "email": "jduzan@vacacoin.com",
24      "name": "Juan",
25      "last_name": "Duran",
26      "role": "Inversionista",
27      "password": "pbkdf2_sha256$260000$2Sy8ujYCapu7cVcn0fA7nZ$2RLSpA+ZyxA/lHr1DuNjClXQtizHJ6hYRz7ieS5+8iA="
28    }
29  }
```

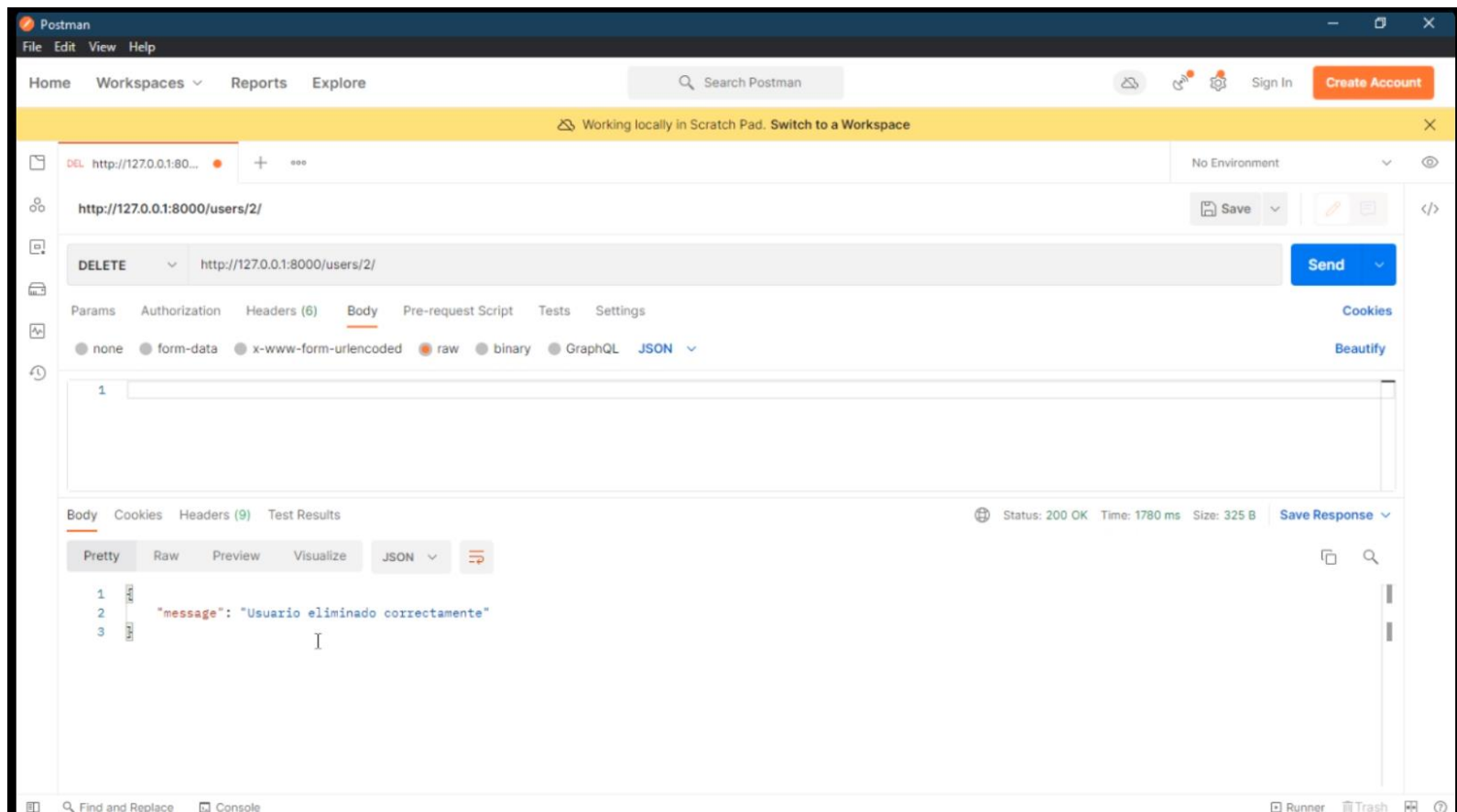
Find and Replace Console

Runner Trash









Postman

File Edit View Help

Home Workspaces Reports Explore

Working locally in Scratch Pad. Switch to a Workspace

GET http://127.0.0.1:8000/users/ No Environment

Save Send

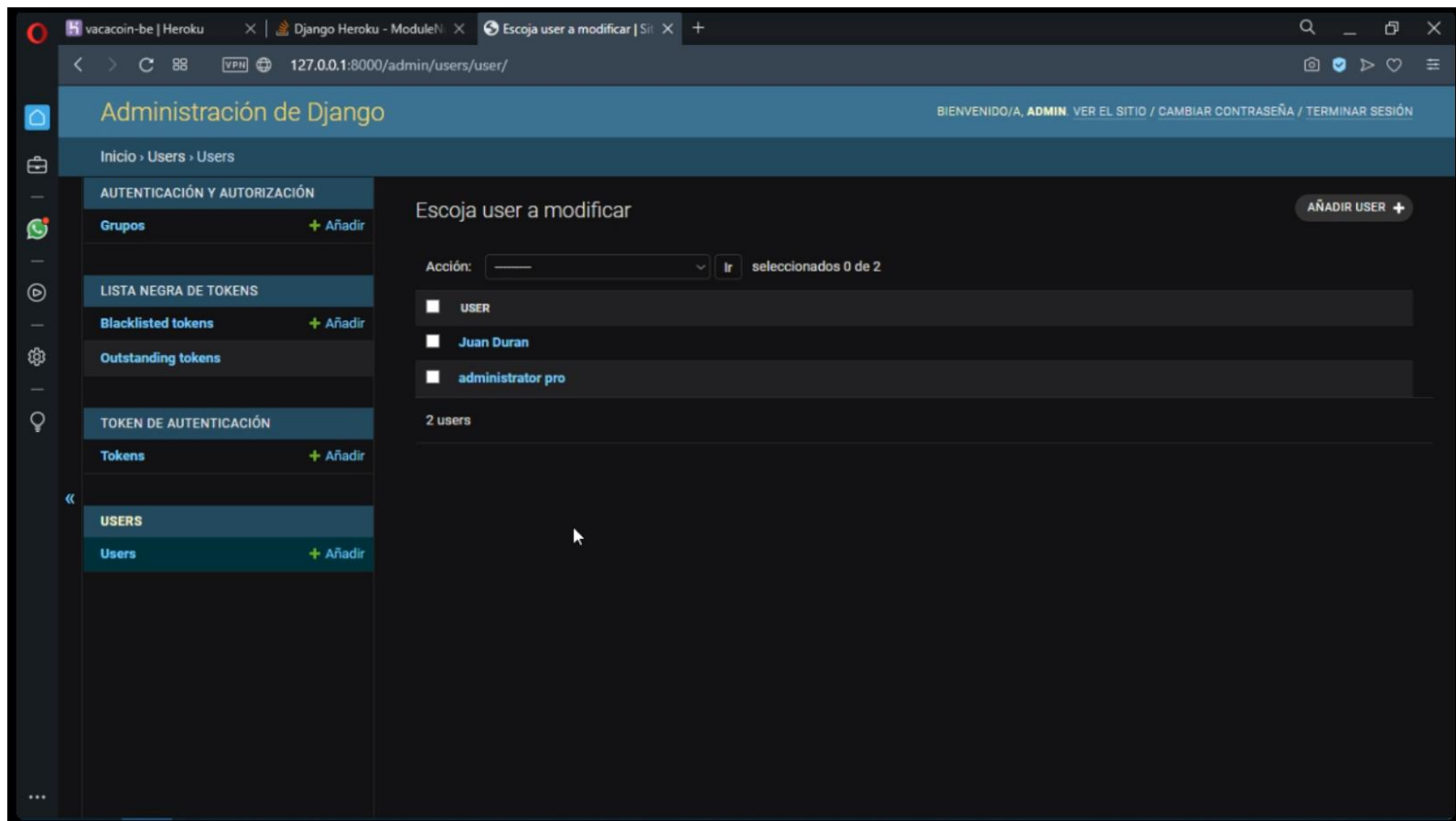
Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

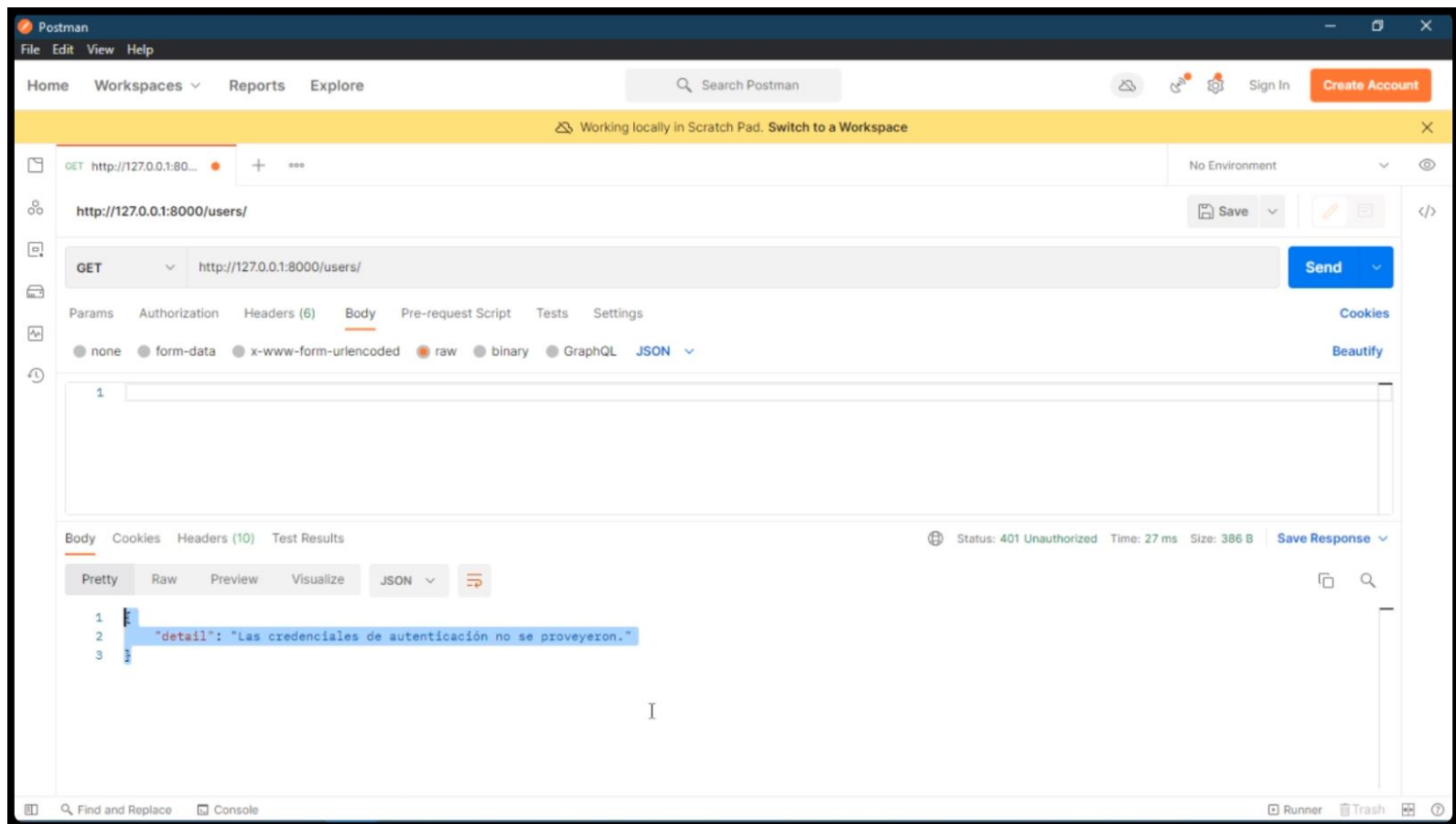
Body Cookies Headers (9) Test Results Status: 200 OK Time: 810 ms Size: 705 B Save Response

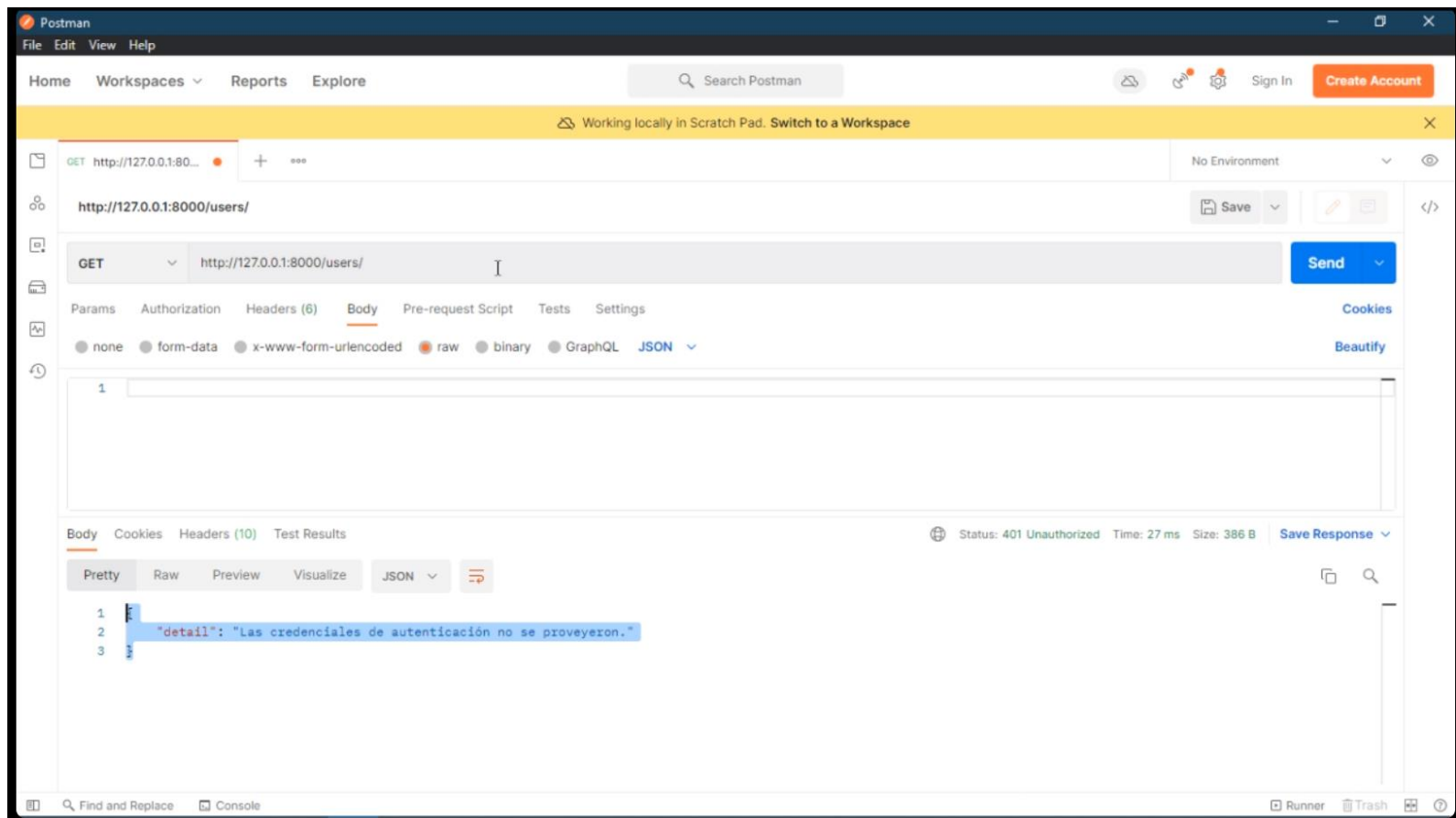
Pretty Raw Preview Visualize JSON

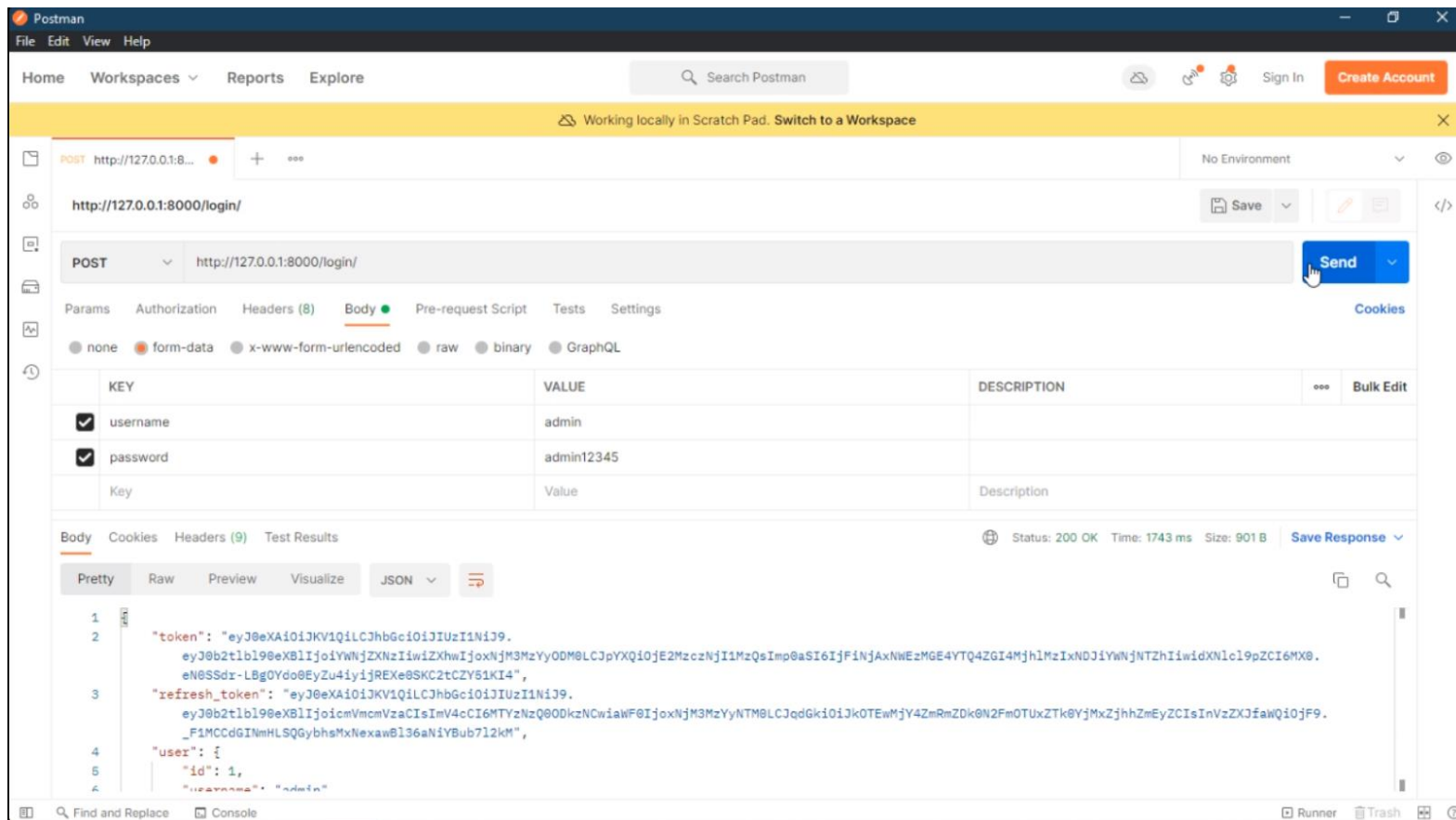
```
1 {
2   "id": 1,
3   "username": "admin",
4   "email": "admin@vacacoin.com",
5   "name": "administrator",
6   "last_name": "pro",
7   "role": null,
8   "password": "pbkdf2_sha256$26000$QxIoURutPtkaWetXkr3Cx$YdBephF32bikDiMOpnykq3BU9ZRuahrPp318SCGDAYM="
9 }
10
11 {
12   "id": 3,
13   "username": "jduran",
14   "email": "jduran@vacacoin.com",
15   "name": "Juan",
16   "last_name": "Duran",
17   "role": "Inversionista",
18   "password": "pbkdf2_sha256$26000$2Sy8ujYCapu7cVcn0fA7nZ$2RLSpA+ZyxA/lHr1DuNjClXQtizHJ6hYRz7ieS5+8iA="
19 }
20 }
```

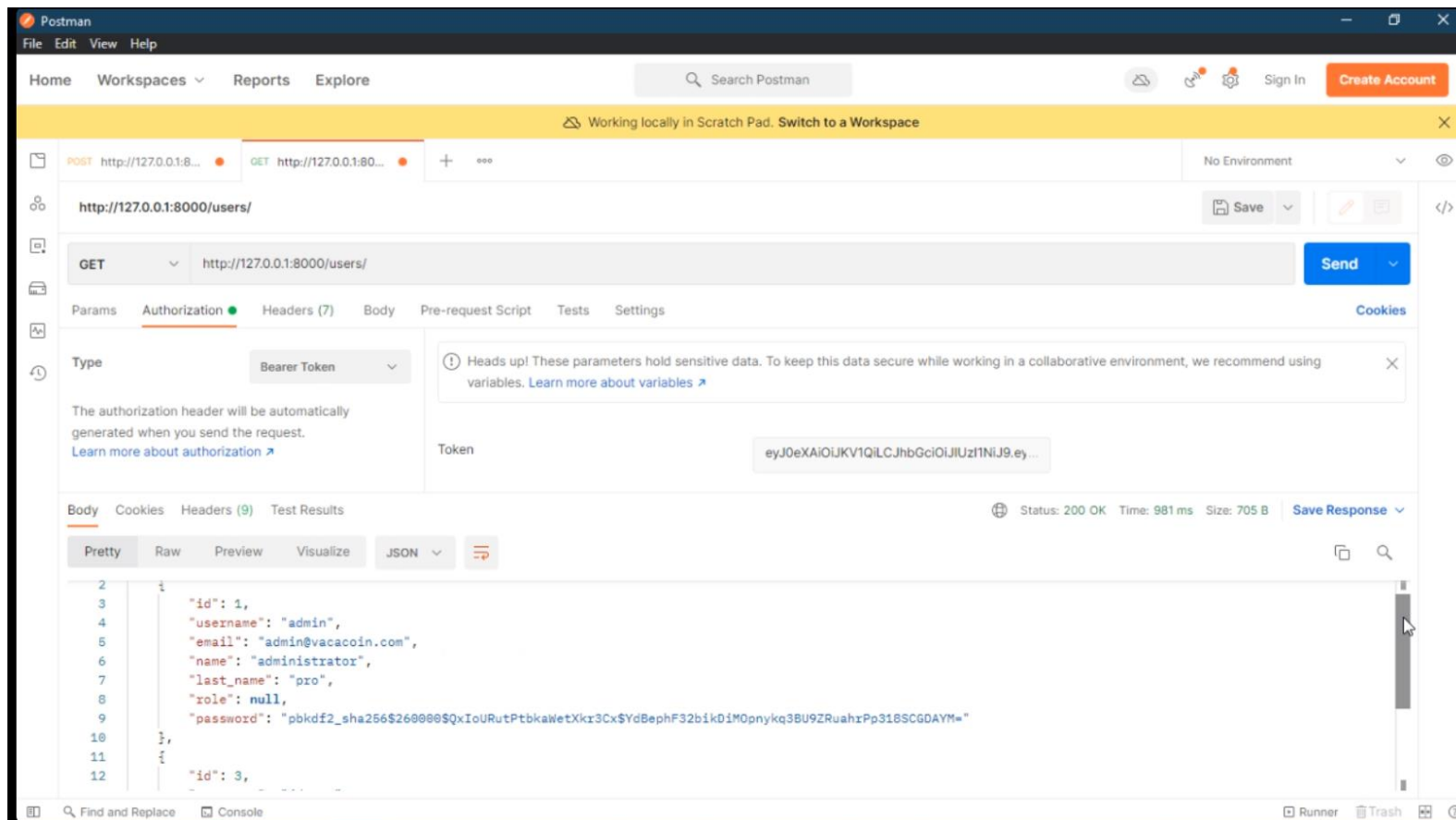
Find and Replace Console Runner Trash

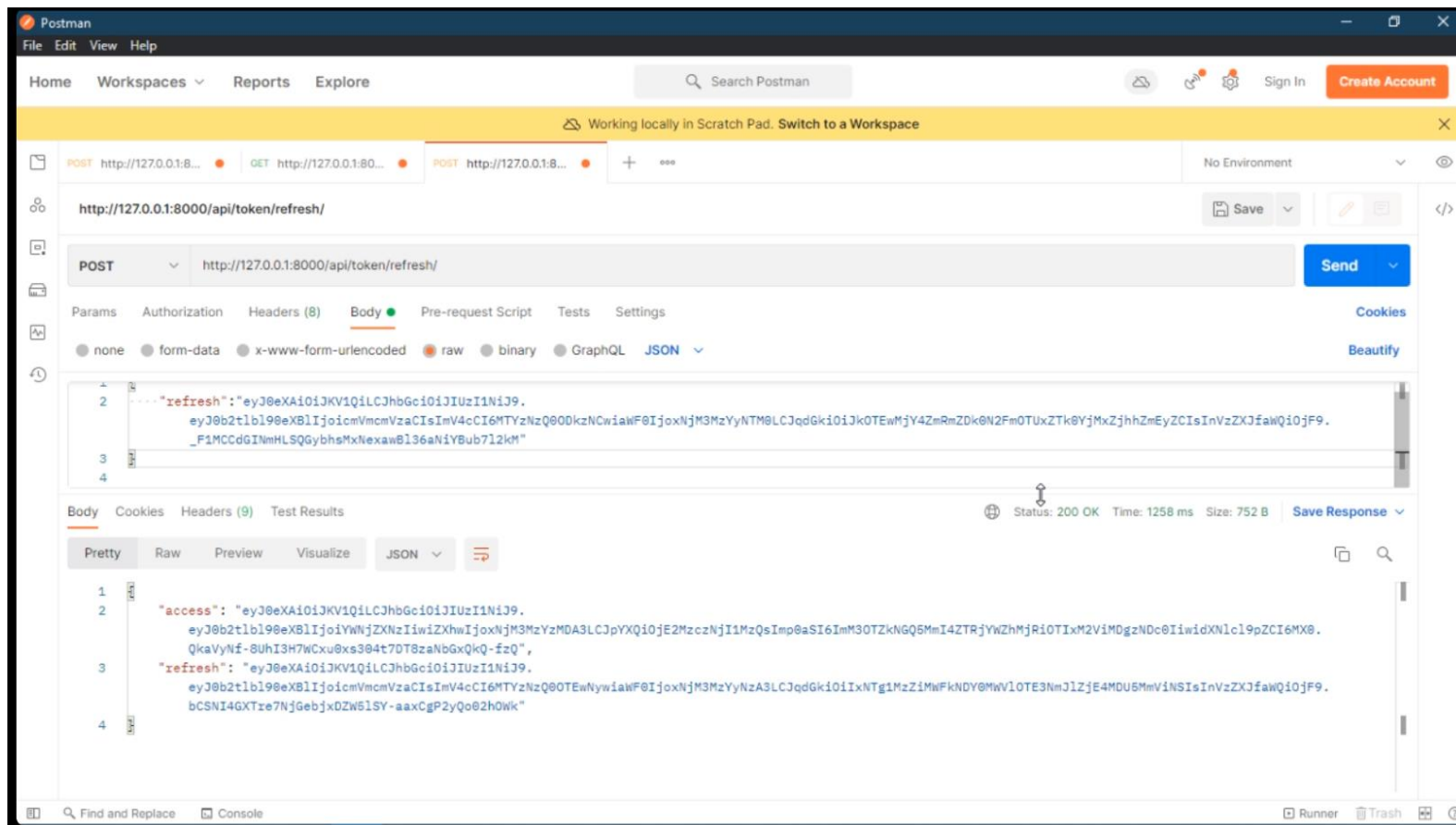




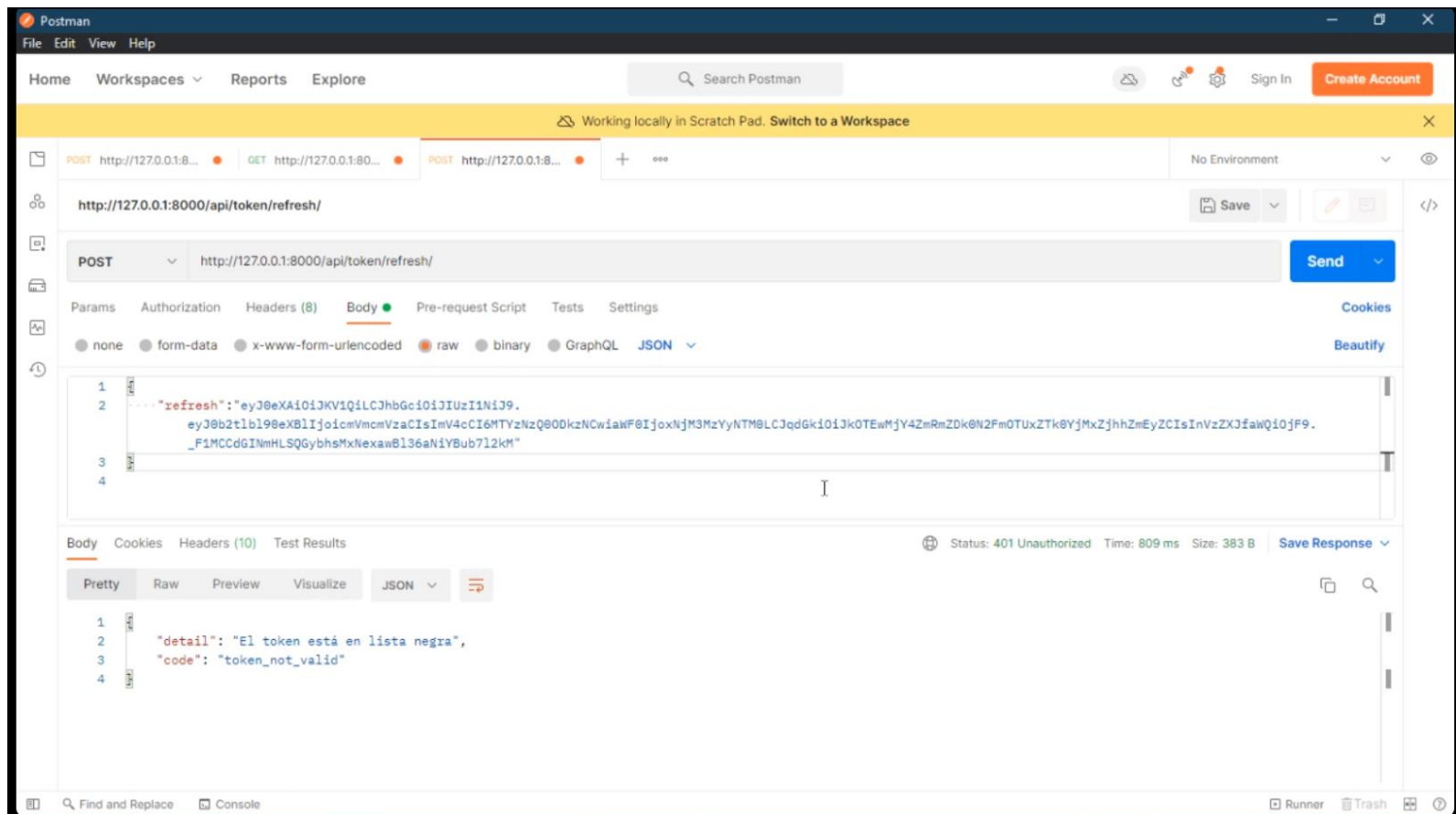














Browser tabs: Django Heroku - ModuleN, Escoja blacklisted token a, Page not found at /, vacacoin-be | Heroku

Address bar: dashboard.heroku.com/apps/vacacoin-be

Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Personal > vacacoin-be

Overview Resources Deploy Metrics Activity Access Settings

Installed add-ons \$0.00/month [Configure Add-ons](#)

Heroku Postgres Hobby Dev postgresql-shaped-78589

Dyno formation \$0.00/month [Configure Dynos](#)

This app is using free dynos

web /bin/sh -c python\ manage.py\ makemigrations\ \&\&\ python... ON

Collaborator activity [Manage Access](#)

daniel.pinto@usantoto.edu.co 1 deploy

Latest activity [All Activity](#)

- daniel.pinto@usantoto.edu.co: Deployed web (c78d0dc2994b) Today at 5:19 PM · v5
- mgogame@gmail.com: @ref:postgresql-shaped-78589 completed provisioning, setting DATABASE\_URL Today at 3:55 PM · v4
- mgogame@gmail.com: Attach DATABASE (@ref:postgresql-shaped-78589) Today at 3:55 PM · v3
- mgogame@gmail.com: Enable Logplex Today at 3:46 PM · v2
- mgogame@gmail.com: Initial release Today at 3:46 PM · v1

heroku.com Blogs Careers Documentation Support Terms of Service Privacy Cookies © 2021 Salesforce.com

Links:

Drive:

<https://drive.google.com/drive/folders/1HnCPo83hi2uD55VZ6AhZRcm0jIDIVUFQ>

Repositorio

<https://github.com/Ravercrown/4a-docs/commit/88728df7c907239e7026853cca55356dd65d522b>