

PANTALLAZOS CODIGO AUTH MS Y USERS.

1. AUTH_MS.

- **URLS.**

```
1  from django.contrib import admin
2  from django.urls import path, include
3
4  urlpatterns = [
5      path('admin/', admin.site.urls),
6      path('rest-auth/', include('rest_auth.urls')), #endpoint Login
7      path('rest-auth/registration/', include('rest_auth.registration.urls')), #endpoint registro
8      path('consultar-usuario/', include('users.urls')) #urls del modulo
9  ]
```

2. USERS.

- **Modelos.**

```
1  from django.db import models
2  from django_countries.fields import CountryField
3
4  # Create your models here.
5
6
7  ITEM_TYPER = (
8      ('Admin', 'Administrador'),
9      ('User', 'Usuario'),
10     ('Inv', 'Invitado'),
11 )
12
13 class Role(models.Model):
14     id = models.AutoField(primary_key = True)
15     type = models.CharField(choices = ITEM_TYPER, max_length = 12)
16
17
18 ITEM_TYPE = (
19     ('CC', 'Cédula de ciudadanía'),
20     ('CE', 'Cédula de extranjería')
21 )
22
23 class User(models.Model):
24
25     id = models.AutoField(primary_key = True)
26     name = models.CharField(max_length = 30, null = False)
27     lastname = models.CharField(max_length = 30, null = False)
28     typeid = models.CharField(choices = ITEM_TYPE, max_length = 10) #tipo de documento
29     numberid = models.CharField(max_length = 12, null = False, blank = True) # numero de cedula
30     phone = models.CharField(max_length = 12, null = False, blank = True)
31     birth = models.DateField()
32     country = CountryField()
33     city = models.CharField(max_length = 20)
34     email = models.EmailField(null = False, unique=True)
35     password = models.CharField(max_length = 12)
36     role = models.ForeignKey(Role, on_delete = models.CASCADE, null = False)
```

- **Vistas.**

```
1 from django.shortcuts import render
2 from rest_framework import generics, authentication, permissions
3 from .models import Role, User
4 from rest_framework.response import Response
5 from .serializer import RoleSerializer, UserSerializer
6
7
8 # Create your views here.
9
10 #CRUD Role
11
12 class RoleListCreate(generics.ListCreateAPIView):
13     queryset = Role.objects.all()
14     serializer_class = RoleSerializer
15
16 class RoleUpdateDelete(generics.RetrieveUpdateDestroyAPIView):
17     queryset = Role.objects.all()
18     serializer_class = RoleSerializer
19
20
21 #CRUD User
22
23 class UserListCreate(generics.ListCreateAPIView):
24     queryset = User.objects.all()
25     serializer_class = UserSerializer
26     authentication_classes = [authentication.TokenAuthentication] #autenticacion de los tokens
27     permission_classes = [permissions.IsAuthenticated]
28
29
30 class UserUpdateDelete(generics.RetrieveUpdateDestroyAPIView):
31     queryset = User.objects.all()
32     serializer_class = UserSerializer
```

- **Serializadores.**

```
1 from rest_framework import serializers
2 from .models import Role, User
3
4 class RoleSerializer(serializers.ModelSerializer): #esta heredando del modelo
5     class Meta:
6         model = Role
7         fields = '__all__'
8
9 class UserSerializer(serializers.ModelSerializer):
10     class Meta:
11         model = User
12         fields = '__all__'
13
```

- **Urls.**

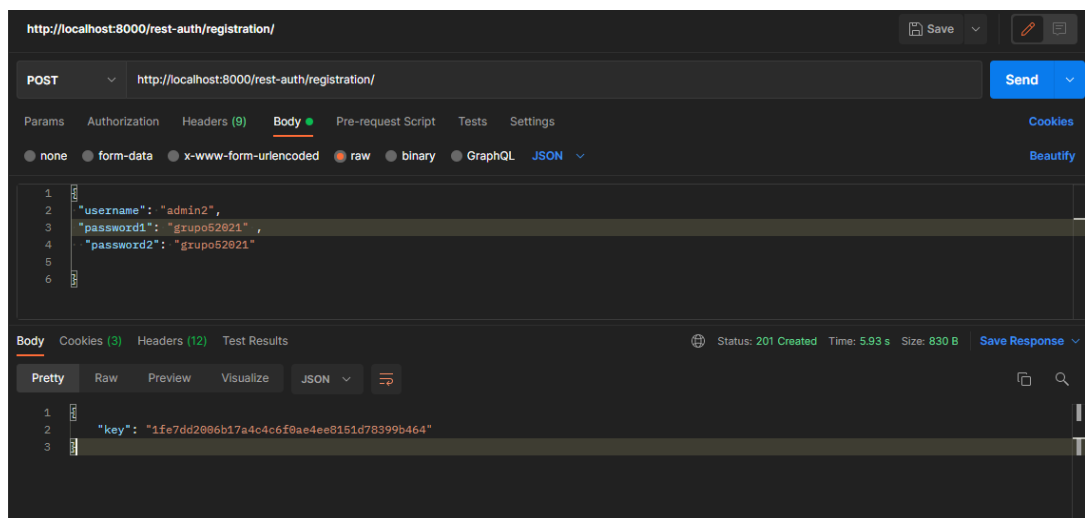
```
1 from django.urls import path
2 from .views import UserListCreate, UserUpdateDelete, RoleListCreate, RoleUpdateDelete
3
4 urlpatterns = [
5     path('rol/' , RoleListCreate.as_view()),
6     path('rol/<pk>/', RoleUpdateDelete.as_view()),
7     path('usuario/', UserListCreate.as_view()),
8     path('usuario/<pk>/', UserUpdateDelete.as_view())
9 ]
```

PRUEBAS POSTMAN – PRIMER MICROSERVICIO PREMIERE.

1. Pruebas del módulo de autenticación y registro.

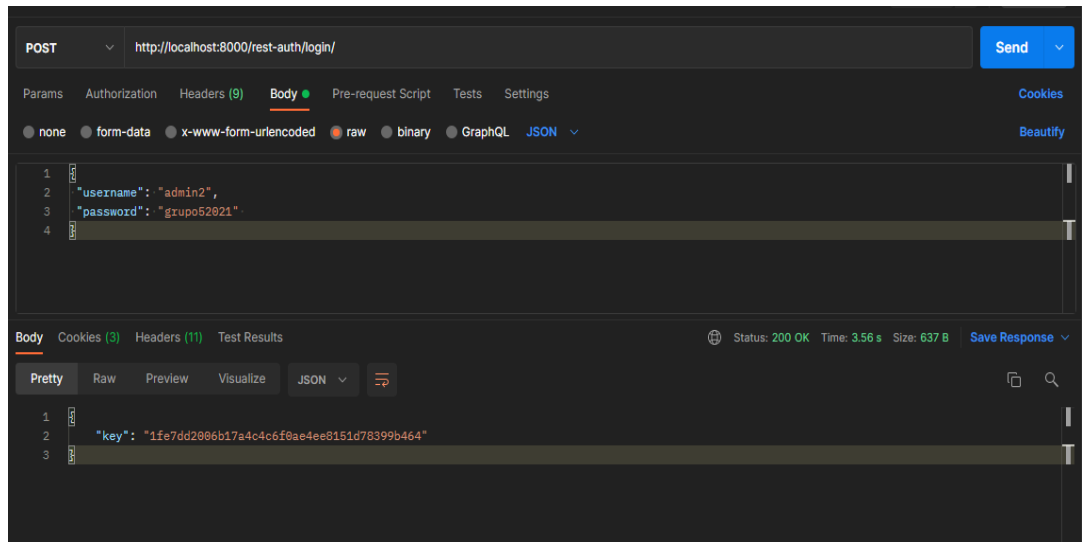
- **Registro.**

Creando un usuario y obteniendo el token respectivo.



- **Login.**

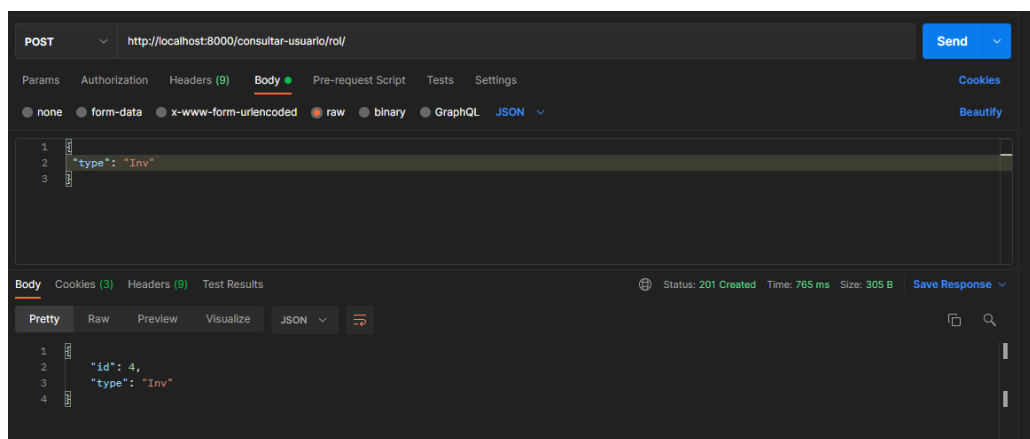
Iniciando sesión con el usuario recientemente creado.



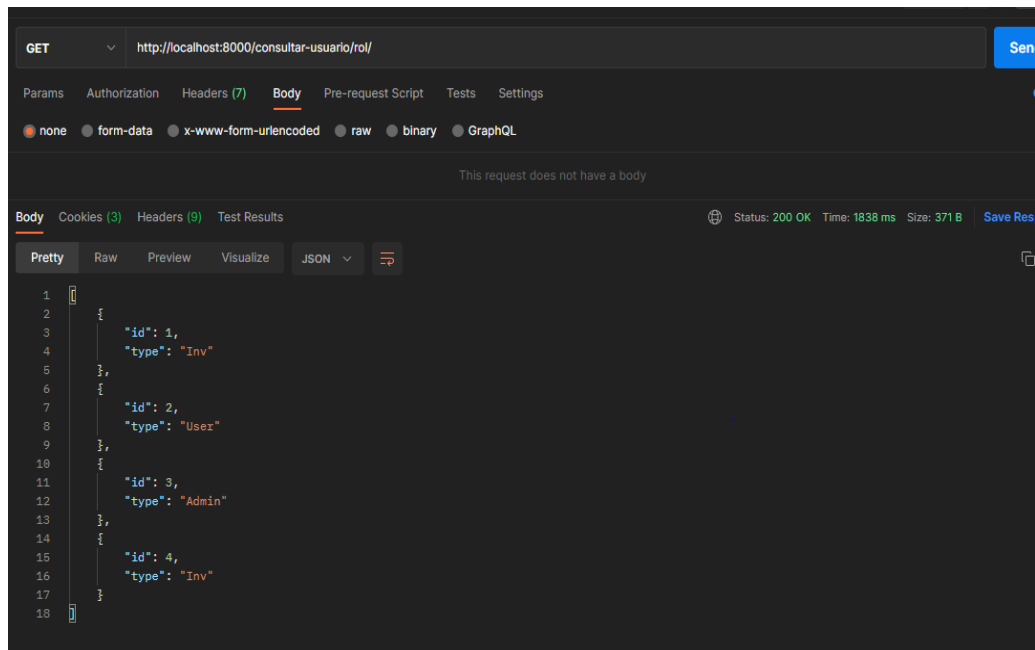
2. Pruebas del módulo de users.

- **Rol.**

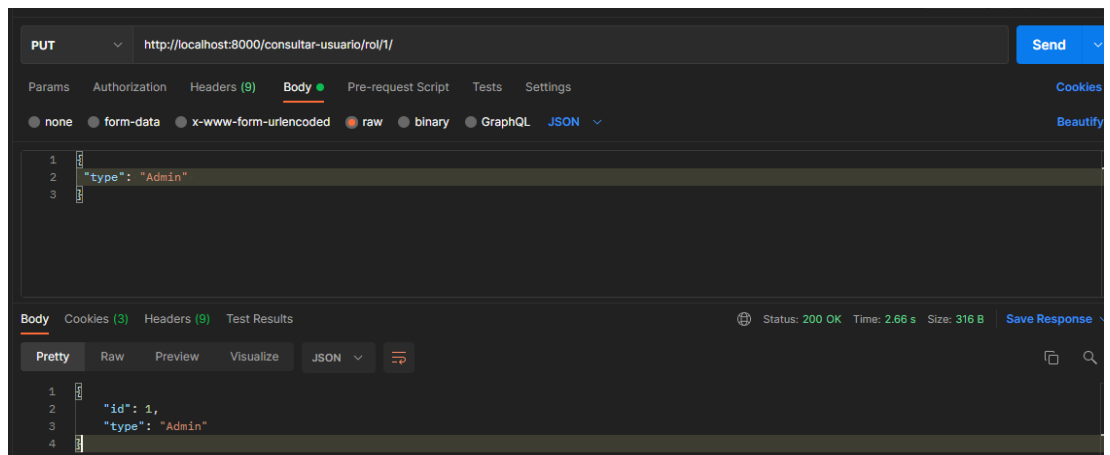
Creando un nuevo rol.



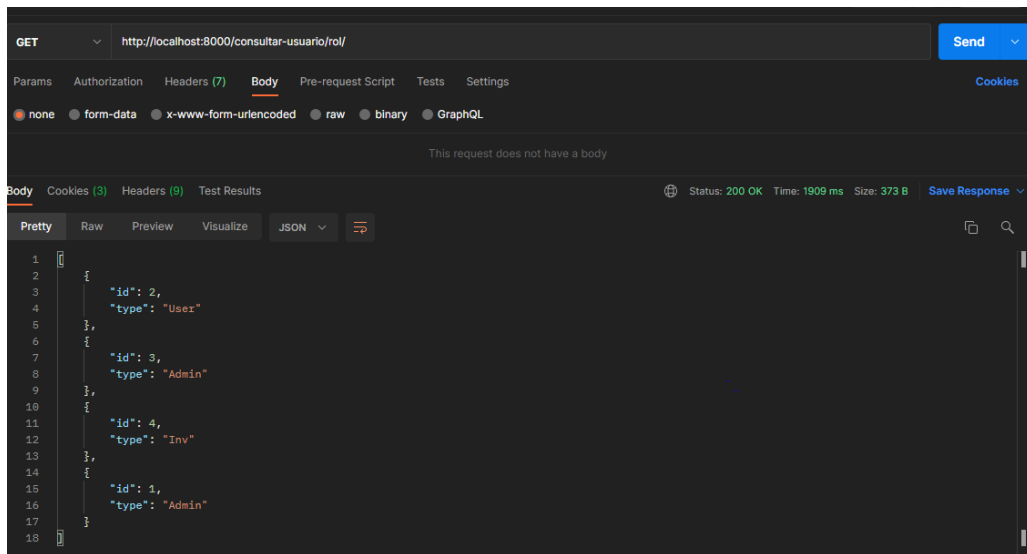
Comprobando la correcta creación del rol recientemente agregado.



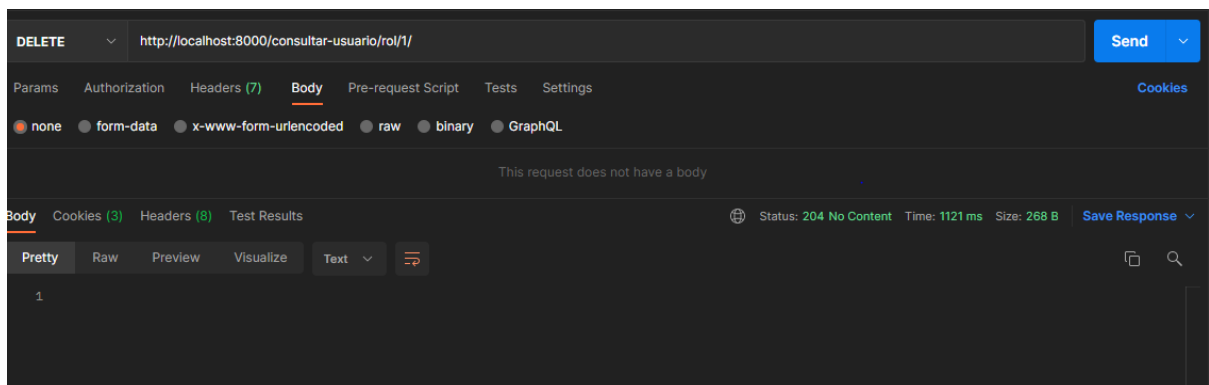
Modificando el tipo del rol que se había creado. Se modificó de tipo invitado a tipo administrador.



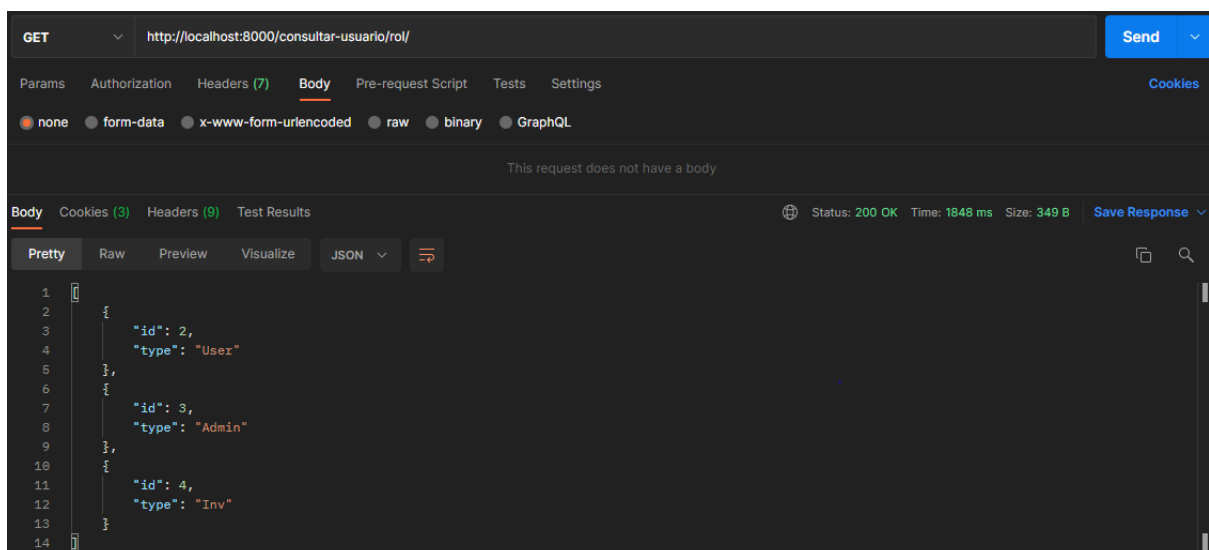
Verificando que efectivamente se hizo el cambio de rol.



Eliminando el rol tipo con el id 1.

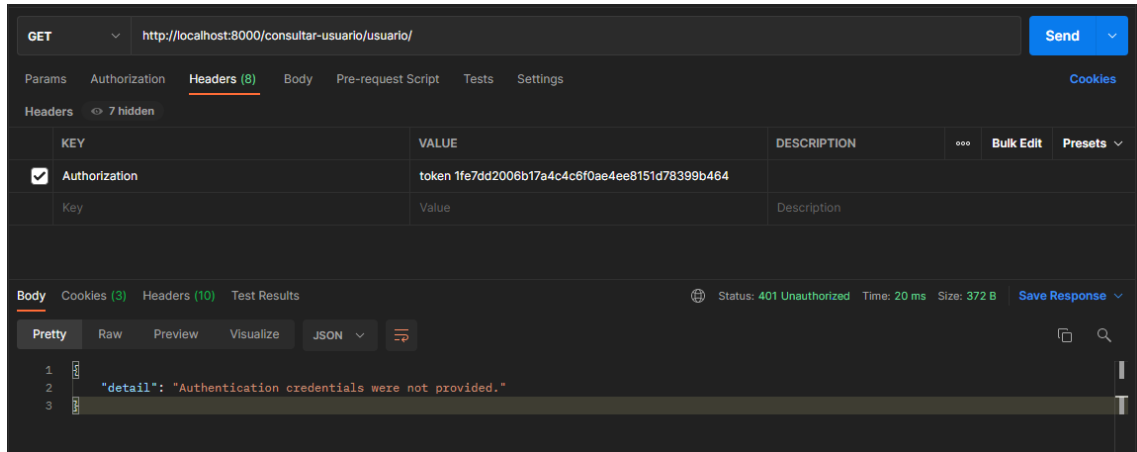


Verificando que el rol con el id 1 se eliminó correctamente.

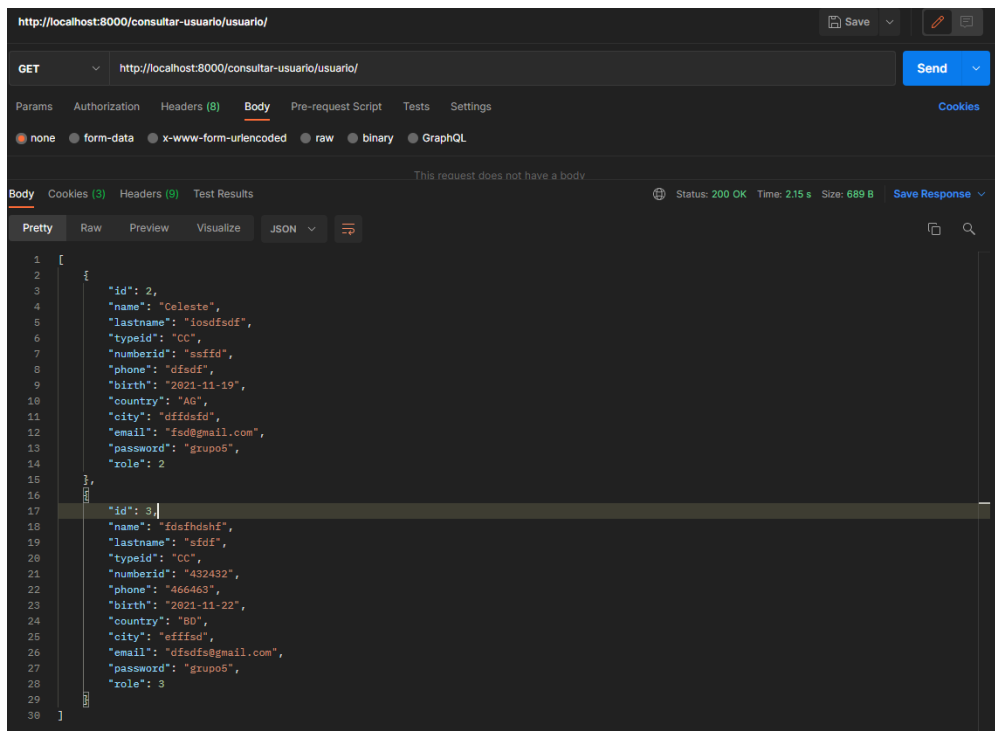


- **User.**

Incluyendo el token de seguridad para poder ver la información detallada de los usuarios.



Consultando los usuarios previamente creados.



Creando un nuevo usuario llamado “Administradora 5”

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/consultar-usuario/usuario/`. The request body is a JSON object representing a new user. The response is a JSON object with an additional `id` field.

```
1 {
2   "name": "Administradora",
3   "lastname": "5",
4   "typeid": "CC",
5   "numberid": "432432",
6   "phone": "466463",
7   "birth": "2021-11-22",
8   "country": "BD",
9   "city": "efffsd",
10  "email": "admin5@gmail.com",
11  "password": "grupo5",
12  "role": 5
13 }
```

Body Cookies (3) Headers (9) Test Results Status: 201 Created Time: 2.52 s Size: 491 B Save Response

```
1 {
2   "id": 4,
3   "name": "Administradora",
4   "lastname": "5",
5   "typeid": "CC",
6   "numberid": "432432",
7   "phone": "466463",
8   "birth": "2021-11-22",
9   "country": "BD",
10  "city": "efffsd",
11  "email": "admin5@gmail.com",
12  "password": "grupo5",
13  "role": 5
14 }
```

Modificando la información del usuario “Administradora 5” a “Administradora Administradora”

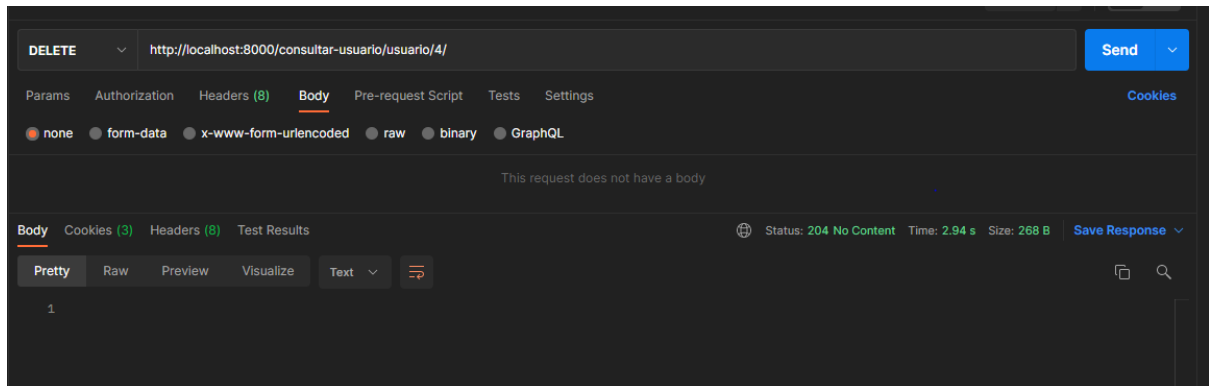
The screenshot shows a REST client interface with a PUT request to `http://localhost:8000/consultar-usuario/usuario/4/`. The request body is a JSON object representing the updated user information. The response is a JSON object with the updated user data.

```
1 {
2   "name": "Administradora",
3   "lastname": "Administradora",
4   "typeid": "CC",
5   "numberid": "432432",
6   "phone": "466463",
7   "birth": "2021-11-22",
8   "country": "BD",
9   "city": "efffsd",
10  "email": "admin5@gmail.com",
11  "password": "grupo5",
12  "role": 5
13 }
```

Body Cookies (3) Headers (9) Test Results Status: 200 OK Time: 3.38 s Size: 513 B Save Response

```
1 {
2   "id": 4,
3   "name": "Administradora",
4   "lastname": "Administradora",
5   "typeid": "CC",
6   "numberid": "432432",
7   "phone": "466463",
8   "birth": "2021-11-22",
9   "country": "BD",
10  "city": "efffsd",
11  "email": "admin5@gmail.com",
12  "password": "grupo5",
13  "role": 5
14 }
```


Eliminando el usuario “Administradora Administradora” asociada a el id 4.



Verificando la eliminación del usuario “Administradora Administradora”

