

Investigación

Registro de nuevos usuarios desde un cliente externo – Investigación

Tema

Registro de nuevos usuarios desde un cliente externo.

Objetivo

Habilitar el registro de usuarios desde herramientas como Postman o desde una interfaz web externa (frontend).

Preguntas de investigación

¿Qué nueva vista deben crear?

Se creó una vista basada en `CreateAPIView` de Django REST Framework:

```
class UserRegisterView(generics.CreateAPIView):  
    serializer_class = UserRegisterSerializer  
    permission_classes = [AllowAny]
```

Permite registrar un nuevo usuario sin autenticación previa.

✓ ¿Qué serializer utilizaron?

Se creó `UserRegisterSerializer`, que hereda de `ModelSerializer` e incluye validaciones:

- Email único.
 - Longitud mínima de contraseña.
-

✓ ¿Cómo se encripta la contraseña?

Se usa el método `create_user()` de Django para asegurar que la contraseña sea guardada con hash:

```
user = User.objects.create_user(  
    username=..., email=..., password=...  
)
```

✓ ¿Qué permisos debe tener la vista?

Se define:

```
permission_classes = [AllowAny]
```

Para que cualquier usuario (aunque no esté autenticado) pueda acceder al endpoint.

✓ ¿Cómo se asigna un grupo automáticamente?

Durante el registro, se obtiene o crea el grupo `'paciente'` y se asocia al nuevo usuario:

```
group, created =  
Group.objects.get_or_create(name='paciente')  
user.groups.add(group)
```

✓ ¿Qué validaciones se incluyen?

- Email debe ser único (`validate_email`).
- Contraseña con mínimo 6 caracteres (`validate_password`).

✓ ¿Qué pruebas se realizaron desde Postman?

1. **POST** a `api/auth/register/` con datos válidos.
2. Confirmación de retorno de tokens `access` y `refresh` .
3. Uso de `access` para obtener perfil desde `api/auth/profile/` .



Resultado del endpoint

- URL: `api/auth/register/`
- Método: `POST`

- Resultado esperado: Usuario creado, tokens devueltos, usuario asignado al grupo `paciente`.
-



Documentación de la API con Swagger (drf-yasg)

Se usó la librería `drf-yasg` para generar documentación interactiva:

Instalación:

```
pip install drf-yasg
```

Configuración en `urls.py`:

```
from drf_yasg.views import get_schema_view
from drf_yasg import openapi
from rest_framework.permissions import AllowAny

schema_view = get_schema_view(
    openapi.Info(
        title="API Hospital",
        default_version='v1',
        description="Documentación interactiva",
    ),
    public=True,
    permission_classes=[AllowAny],
)

urlpatterns += [
    path('swagger/', schema_view.with_ui('swagger',
cache_timeout=0), name='schema-swagger-ui'),
    path('redoc/', schema_view.with_ui('redoc',
```

```
cache_timeout=0), name='schema-redoc'),  
]
```



Capturas del Cliente Web

url: `http://127.0.0.1:8000/login/`

1. Login de usuario

[Log in](#) ☐ [Sign up](#)

Log in

2. Registro de nuevo usuario (Sign up)

Log in



Sign up

Sign up as Patient

Confirm!

View Profile

3. Visualización de datos de perfil – usuario "Sara"

Profile Data

Username: sara
Email: svelez@a.com
First Name: sara
Last Name: velez
Active: Yes
Staff: No
Superuser: No
Groups: paciente
Date Joined: 9/5/2025, 19:46:47
Last Login: N/A

Close

Confirm!

4. Visualización de datos de perfil – usuario "Allan"

Profile Data

Username: kromm

Email: allan@gmail.com

First Name: allan

Last Name: velez

Active: Yes

Staff: No

Superuser: No

Groups: paciente

Date Joined: 9/5/2025, 19:28:23

Last Login: N/A

Close

5. Registro de nuevo usuario – Formulario completo

Log in



Sign up

Sign up as Pacient

sara

svelez@a.com

sara

velez

.....

Confirm!

View Profile

6. Error de registro mostrado en modal

Log in



[Sign up](#)



Registration failed

Please try again.

OK

.....

Confirm!

View Profile



Conclusión

Se logró habilitar un endpoint funcional para registro desde el frontend, incluyendo validaciones, asignación automática de grupos, y autenticación JWT. El sistema es fácilmente extensible y seguro para el manejo de usuarios.

Autor: Allan José Vélez González – FWD Costa Rica

Fecha de entrega: Lunes 12 de mayo de 2025