Documentación API de Usuarios

Descripción General

Esta API proporciona un sistema completo de autenticación y gestión de usuarios con características avanzadas de seguridad.

Características de Seguridad Implementadas

1. Autenticación

- JWT (JSON Web Tokens) para autenticación sin estado
- Tokens de acceso y refresco
- Blacklisting de tokens para logout seguro
- Verificación de cuentas activas

2. Rate Limiting

- Login: 5 intentos por minuto
- Reset de contraseña por email: 3 intentos por hora
- Rate limiting general para usuarios autenticados

3. Protección contra Ataques

- Bloqueo temporal después de 5 intentos fallidos de login
- Tokens seguros para recuperación de contraseña
- Expiración de tokens de recuperación (24 horas)
- Validación de contraseñas seguras
- Protección CSRF en todas las peticiones

4. Logging y Monitoreo

- Registro de intentos de login fallidos
- Registro de cambios de contraseña
- Registro de creación de usuarios
- Monitoreo de actividades sospechosas

Endpoints

Autenticación

POST /usuarios/login/

Login de usuario

```
{
    "email": "usuario@ejemplo.com",
    "password": "contraseña"
}
```

Respuesta:

```
{
    "refresh": "token_refresco",
    "access": "token_acceso",
    "user": {
        "id": 1,
        "email": "usuario@ejemplo.com",
        "username": "usuario",
        ...
    }
}
```

GET /usuarios/status/

Verificar estado de autenticación

```
{
    "is_authenticated": true,
    "user": {
        "id": 1,
        "email": "usuario@ejemplo.com",
        ...
    }
}
```

POST /usuarios/logout/

Cerrar sesión (requiere autenticación)

```
{
    "message": "Logout exitoso"
}
```

Gestión de Usuarios

POST /usuarios/

Registro de nuevo usuario

```
{
    "email": "nuevo@ejemplo.com",
    "username": "nuevo_usuario",
    "password": "contraseña123",
    "password2": "contraseña123",
    "first_name": "Nombre",
    "last_name": "Apellido",
    "telefono": "123456789"
}
```

GET /usuarios/me/

Obtener perfil del usuario actual (requiere autenticación)

```
{
    "id": 1,
```

```
    "email": "usuario@ejemplo.com",
    "username": "usuario",
    "first_name": "Nombre",
    "last_name": "Apellido",
    "telefono": "123456789",
    "fecha_registro": "2025-01-27T10:00:00Z"
}
```

Gestión de Contraseñas

POST /usuarios/solicitar_reset_password/

Solicitar reset de contraseña

```
{
    "email": "usuario@ejemplo.com"
}
```

POST /usuarios/reset_password_confirm/

Confirmar reset de contraseña

```
{
    "token": "token_recibido_por_email",
    "new_password": "nueva_contraseña",
    "new_password2": "nueva_contraseña"
}
```

POST /usuarios/cambiar_password/

Cambiar contraseña (requiere autenticación)

```
{
    "old_password": "contraseña_actual",
    "new_password": "nueva_contraseña",
    "new_password2": "nueva_contraseña"
}
```

Códigos de Estado

- 200: Operación exitosa
- 201: Recurso creado exitosamente
- 400: Error en la solicitud
- 401: No autorizado
- 403: Prohibido
- 404: No encontrado
- 429: Demasiadas solicitudes
- 500: Error interno del servidor

Recomendaciones de Seguridad

1. **Almacenamiento de Tokens**
    - Almacenar tokens JWT de forma segura

- No almacenar en localStorage por riesgo de XSS
- Preferir almacenamiento en cookies httpOnly

2 **Manejo de Contraseñas**

- Usar contraseñas fuertes (mínimo 8 caracteres)
- Combinar mayúsculas, minúsculas, números y símbolos
- No reutilizar contraseñas

3 **Comunicación**

- Usar HTTPS en producción
- Configurar CORS apropiadamente
- Implementar certificados SSL válidos

4 **Configuración en Producción**

- Deshabilitar DEBUG
- Configurar ALLOWED_HOSTS
- Usar backend SMTP real para emails
- Configurar logging apropiado