

# Usuarios de Prueba - IoT Sensor Platform

---

Este documento contiene la información de los usuarios de prueba creados por el comando `crear_datos_prueba`.

---

## Credenciales de Usuario









---

### Superusuario (Administrador)

**Creado con:** `python manage.py crear_superuser`

- **Username:** `admin`
- **Password:** `admin123`
- **Email:** `admin@iotplatform.com`
- **Rol:** Superusuario
- **Permisos:** Acceso completo a todo el sistema

#### Funcionalidades:

-  Gestionar usuarios, roles y permisos
  -  Crear, editar y eliminar sensores
  -  Crear, editar y eliminar dispositivos
  -  Asignar sensores a dispositivos
  -  Asignar operadores a dispositivos
  -  Ver todas las lecturas
  -  Crear lecturas
  -  Ver dashboard completo
- 

### Operador 1






**Creado con:** `python manage.py crear_datos_prueba`

- **Username:** `operador1`
- **Password:** `operador123`
- **Email:** `operador1@iotplatform.com`
- **Nombre:** Operador Uno
- **Rol:** Operador
- **Tipo:** Usuario Interno

#### Dispositivos Asignados:

- Raspberry Pi - Sala

#### Funcionalidades:

-  Ver todos los sensores
-  Crear y gestionar sensores
-  Ver todos los dispositivos
-  Gestionar sus dispositivos asignados
-  Asignar sensores a sus dispositivos

- ☒ Crear lecturas para sus dispositivos
  - ☒ Ver lecturas de sus dispositivos
  - ☒ Ver dashboard con sus estadísticas
- 

## Operador 2

**Creado con:** `python manage.py crear_datos_prueba`

- **Username:** operador2
- **Password:** operador123
- **Email:** operador2@iotplatform.com
- **Nombre:** Operador Dos
- **Rol:** Operador
- **Tipo:** Usuario Interno

**Dispositivos Asignados:**

- ESP32 - Cocina

**Funcionalidades:** (Iguales a Operador 1)

---

## Operador 3

**Creado con:** `python manage.py crear_datos_prueba`

- **Username:** operador3
- **Password:** operador123
- **Email:** operador3@iotplatform.com
- **Nombre:** Operador Tres
- **Rol:** Operador
- **Tipo:** Usuario Interno

**Dispositivos Asignados:**

- Arduino - Jardín

**Funcionalidades:** (Iguales a Operador 1)

---

## Usuario de Solo Lectura

**Creado con:** `python manage.py crear_datos_prueba`

- **Username:** viewer
- **Password:** viewer123
- **Email:** viewer@iotplatform.com
- **Nombre:** Usuario Lectura
- **Rol:** Solo Lectura
- **Tipo:** Usuario Externo

**Funcionalidades:**

- ☒ Ver lista de usuarios

- ☒ Ver sensores
  - ☒ Ver dispositivos
  - ☒ Ver lecturas
  - ☒ Ver dashboard
  - ☒ No puede crear, editar o eliminar nada
- 



## Datos de Prueba Creados

---

### Sensores (6 en total)

#### 1. DHT22 - Temperatura

- Tipo: Temperatura
- Unidad: °C
- Rango: -40 a 80

#### 2. DHT22 - Humedad

- Tipo: Humedad
- Unidad: %
- Rango: 0 a 100

#### 3. BMP280 - Presión

- Tipo: Presión
- Unidad: hPa
- Rango: 300 a 1100

#### 4. LDR - Luz

- Tipo: Luz
- Unidad: lux
- Rango: 0 a 1000

#### 5. PIR - Movimiento

- Tipo: Movimiento
- Unidad: boolean
- Rango: 0 a 1

#### 6. MQ-2 - Gas

- Tipo: Gas
  - Unidad: ppm
  - Rango: 0 a 10000
- 

### Dispositivos (4 en total)

#### 1. Raspberry Pi - Sala

- Tipo: Raspberry Pi
- Identificador: RPI-SALA-001
- Ubicación: Sala Principal
- Operador: operador1
- Sensores asignados: 3 (aleatorio)

## 2. ESP32 - Cocina

- Tipo: ESP32
- Identificador: ESP32-COCINA-001
- Ubicación: Cocina
- Operador: operador2
- Sensores asignados: 3 (aleatorio)

## 3. Arduino - Jardín

- Tipo: Arduino
- Identificador: ARD-JARDIN-001
- Ubicación: Jardín Trasero
- Operador: operador3
- Sensores asignados: 3 (aleatorio)

## 4. ESP32 - Habitación

- Tipo: ESP32
- Identificador: ESP32-HAB-001
- Ubicación: Habitación Principal
- Operador: operador1
- Sensores asignados: 3 (aleatorio)

---

## Lecturas

- **Total:** ~120 lecturas
- **Distribución:** 10 lecturas por cada sensor asignado a cada dispositivo
- **Valores:** Aleatorios dentro del rango permitido de cada sensor



## Pruebas de Autenticación

### Login con cURL

#### Admin

```
curl -X POST http://localhost:8000/api/auth/login/ \  
-H "Content-Type: application/json" \  
-d '{  
  "username": "admin",  
  "password": "admin123"  
}
```

#### Operador

```
curl -X POST http://localhost:8000/api/auth/login/ \  
-H "Content-Type: application/json" \  
-d '{  
  "username": "operador1",  
  "password": "operador123"  
}
```

## Solo Lectura

```
curl -X POST http://localhost:8000/api/auth/login/ \
-H "Content-Type: application/json" \
-d '{
  "username": "viewer",
  "password": "viewer123"
}'
```

### Respuesta exitosa:

```
{
  "user": { "..."},
  "refresh": "eyJ0eXAiOiJKV1QiLCJhbGc...",
  "access": "eyJ0eXAiOiJKV1QiLCJhbGc...",
  "message": "Login exitoso"
}
```



## Casos de Prueba

### Como Superusuario (admin)

1. ☒ Crear un nuevo usuario
2. ☒ Crear un nuevo sensor
3. ☒ Crear un nuevo dispositivo
4. ☒ Asignar sensor a dispositivo
5. ☒ Asignar operador a dispositivo
6. ☒ Crear lectura
7. ☒ Ver todas las lecturas
8. ☒ Desactivar usuario

### Como Operador (operador1)

1. ☒ Ver sus dispositivos asignados
2. ☒ Crear un nuevo sensor
3. ☒ Asignar sensor a su dispositivo
4. ☒ Crear lectura para su dispositivo
5. ☒ Ver lecturas de sus dispositivos
6. ☒ NO puede ver dispositivos de otros operadores
7. ☒ NO puede asignar operadores
8. ☒ NO puede crear usuarios

### Como Solo Lectura (viewer)

1. ☒ Ver lista de sensores
2. ☒ Ver lista de dispositivos
3. ☒ Ver lecturas
4. ☒ Ver estadísticas del dashboard
5. ☒ NO puede crear sensores
6. ☒ NO puede crear dispositivos

7. ❌ NO puede crear lecturas
8. ❌ NO puede modificar nada



## Endpoints para Probar

### Públicos (Sin autenticación)

```
POST /api/auth/register/
POST /api/auth/login/
POST /api/auth/refresh/
```

### Requieren Autenticación

```
# Header necesario: Authorization: Bearer {access_token}

GET /api/users/me/
GET /api/sensors/
POST /api/sensors/
GET /api/devices/
POST /api/devices/
POST /api/devices/1/assign-sensor/
GET /api/readings/
POST /api/readings/
GET /api/dashboard/stats/
```



## Resetear Datos de Prueba

Si quieres resetear los datos de prueba:

### Con Docker:

```
docker-compose down -v # Elimina volúmenes
docker-compose up --build -d
docker-compose exec django python manage.py crear_superuser
docker-compose exec django python manage.py crear_datos_prueba
```

### Sin Docker:

```
# Eliminar base de datos
sudo -u postgres psql -c "DROP DATABASE iot_sensor_db;"
sudo -u postgres psql -c "CREATE DATABASE iot_sensor_db;"
sudo -u postgres psql -c "GRANT ALL PRIVILEGES ON DATABASE iot_sensor_db TO iot_user;"

# Recrear estructura
python manage.py makemigrations
python manage.py migrate
python manage.py crear_permisos_default
python manage.py crear_roles_default
python manage.py crear_superuser
python manage.py crear_datos_prueba
```

---

## Notas Importantes

1. **Cambiar Contraseñas en Producción:** Las contraseñas de estos usuarios son de prueba y deben ser cambiadas en producción.
  2. **Tokens JWT:** Los tokens de acceso expiran en 60 minutos. Usa el refresh token para obtener un nuevo access token.
  3. **Permisos por Rol:** Los operadores solo pueden ver y gestionar sus propios dispositivos y lecturas relacionadas.
  4. **Datos Aleatorios:** Las lecturas generadas tienen valores aleatorios dentro del rango permitido de cada sensor.
- 

## Verificar Datos Creados

### Desde Django Shell:

```
python manage.py shell
```

```
from api.models import CustomUser, Sensor, Dispositivo, Lectura

# Contar usuarios
print(f"Usuarios: {CustomUser.objects.count()}")

# Contar sensores
print(f"Sensores: {Sensor.objects.count()}")

# Contar dispositivos
print(f"Dispositivos: {Dispositivo.objects.count()}")

# Contar lecturas
print(f"Lecturas: {Lectura.objects.count()}")

# Ver dispositivos del operador1
operador1 = CustomUser.objects.get(username='operador1')
dispositivos = Dispositivo.objects.filter(operador_asignado=operador1)
print(f"Dispositivos de operador1: {dispositivos.count()}")
for d in dispositivos:
    print(f"    - {d.nombre}")
```

---

¡Usa estos usuarios para explorar todas las funcionalidades de la plataforma! 🚀