

# Manual de Uso:

# Sistema de Login con FastAPI, JWT y

# API Key

Versión 1.0.0

## Introducción

Este manual describe los pasos para instalar, configurar y probar un sistema de autenticación y autorización básico, desarrollado con **FastAPI**, **JWT** y un archivo JSON como almacenamiento de usuarios. También explica cómo usar **API Key** para asegurar los endpoints de *registro* y *login*, y cómo proteger las rutas privadas con tokens JWT (por ejemplo, */user/me*).

El objetivo de este proyecto es servir como *guía educativa* y tutorial para quienes deseen aprender los fundamentos de la seguridad en APIs con Python y FastAPI. No se recomienda su uso en producción, ya que no implementa cifrado de contraseñas ni otras prácticas avanzadas.

## Estructura del Proyecto

```
basiclogin/  
  main.py  
  users.json  
  requirements.txt  
  venv/ (opcional si usas entorno virtual)
```

## Archivos principales

- **main.py**: Contiene la aplicación FastAPI y la lógica de autenticación.
- **users.json**: Archivo JSON usado como “base de datos” para los usuarios.
- **requirements.txt**: Lista de dependencias necesarias para ejecutar el proyecto.

# Instalación y Ejecución

1. Crear entorno virtual (opcional, pero recomendado):

```
python3 -m venv venv
source venv/bin/activate # Linux/Mac
```

2. Instalar dependencias:

```
pip install -r requirements.txt
```

3. Ejecutar la aplicación:

```
uvicorn main:app --reload
```

La API se iniciará en `http://127.0.0.1:8000`.

## Endpoints y Seguridad

### /register (POST)

- **Protegido por API Key:** Se requiere cabecera `X-API-KEY: MI_API_KEY_SECRETA`.
- **Cuerpo (JSON):**

```
{
  "username": "alice",
  "password": "alicepass"
}
```

- **Objetivo:** Registra un nuevo usuario en el archivo `users.json`.

### /login (POST)

- **Protegido por API Key:** Se requiere `X-API-KEY`.
- **Cuerpo (JSON):**

```
{
  "username": "alice",
  "password": "alicepass"
}
```

- **Objetivo:** Autentica al usuario y devuelve un token **JWT**.

### /user/me (GET)

- **Protegido por token JWT:** Se requiere cabecera `Authorization: Bearer <access_token>`.
- **Objetivo:** Devuelve la información del usuario extraída del token JWT.

## Pruebas en SoapUI

**SoapUI** es una herramienta creada originalmente para servicios SOAP, pero también puede usarse para probar APIs REST.

### Configuración de Proyecto REST

1. **Crear nuevo proyecto REST:** Menú File >New REST Project.
2. **URL inicial:**

```
http://127.0.0.1:8000/register
```

3. SoapUI generará un proyecto con una petición REST inicial que puedes ajustar.

### Registrar Usuario (/register)

- **Method:** POST
- **Endpoint:** `http://127.0.0.1:8000/register`
- **Headers:**

```
X-API-KEY: MI_API_KEY_SECRETA
Content-Type: application/json
```

- **Body (JSON):**

```
{
  "username": "alice",
  "password": "alicepass"
}
```

## Login de Usuario (/login)

- Method: POST
- Endpoint: `http://127.0.0.1:8000/login`
- Headers:

```
X-API-KEY: MI_API_KEY_SECRETA
Content-Type: application/json
```

- Body (JSON):

```
{
  "username": "alice",
  "password": "alicepass"
}
```

## Obtener Datos de Usuario Actual (/user/me)

- Method: GET
- Endpoint: `http://127.0.0.1:8000/user/me`
- Headers:

```
Authorization: Bearer <access_token>
```

## Notas de Seguridad y Mejoras

- **Hasheo de contraseñas:** Actualmente, el sistema almacena contraseñas en texto plano. Para producción, se recomienda usar librerías como `bcrypt` o `passlib`.

- **Cambio de API Key y SECRET\_KEY:** Guardar estas claves en variables de entorno.
- **Refresco de Tokens (Refresh Tokens):** Implementación opcional para extender la vida de los JWT sin volver a pedir credenciales.
- **Base de Datos real:** Migrar a SQLite, PostgreSQL o cualquier otro sistema para un proyecto más escalable.

## Conclusión

Este proyecto sirve como introducción a la autenticación con FastAPI, JWT y protección por API Key. Proporciona las bases para que puedas extenderlo y adecuarlo a necesidades más avanzadas. Puedes publicar el repositorio en GitHub y compartirlo en LinkedIn como ejemplo de tutorial o prueba de concepto.