

Crear una aplicación web rápida en Python utilizando el framework FastAPI

Instalación de FastAPI

```
# Primero, instala FastAPI y Uvicorn usando pip
# pip install fastapi uvicorn
```

Crear una API simple

```
from fastapi import FastAPI

# Crear una instancia de FastAPI
app = FastAPI()

# Definir una ruta simple
@app.get("/")
def read_root():
    return {"Hello": "World"}

# Definir una ruta con parámetros
@app.get("/items/{item_id}")
def read_item(item_id: int, q: str = None):
    return {"item_id": item_id, "q": q}
```

Ejecutar el servidor

```
# Guarda el archivo anterior como main.py y ejecuta el
servidor usando Uvicorn
# uvicorn main:app --reload
```

Usar parámetros de consulta y rutas

```
from fastapi import FastAPI
from typing import Optional

app = FastAPI()

@app.get("/items/{item_id}")
def read_item(item_id: int, q: Optional[str] = None):
    return {"item_id": item_id, "q": q}

@app.get("/users/{user_id}/items/{item_id}")
def read_user_item(user_id: int, item_id: int):
    return {"user_id": user_id, "item_id": item_id}
```

Crear y usar modelos con Pydantic

```
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

class Item(BaseModel):
    name: str
    description: str = None
    price: float
    tax: float = None

@app.post("/items/")
def create_item(item: Item):
    return {"item": item}
```

Manejo de errores

```
from fastapi import FastAPI, HTTPException

app = FastAPI()

@app.get("/divide/{x}/{y}")
def divide(x: int, y: int):
    if y == 0:
        raise HTTPException(status_code=400, detail="Division
by zero is not allowed")
    return {"result": x / y}
```