



**Universidad  
Gerardo Barrios**

**UNIVERSIDAD GERARDO BARRIOS, SAN MIGUEL.  
FACULTAD DE CIENCIA Y TECNOLOGÍA.**

**CÁTEDRA:  
PROGRAMACIÓN COMPUTACIONAL III**

**CATEDRÁTICO:  
WILLIAN MONTES**

**ACTIVIDAD:  
PARCIAL SEGUNDO**

**ESTUDIANTES:  
JULIO ALBERTO FUNES ALFARO  
SHELSY RUBI MELENDEZ RODRIGUEZ**

## INDICE

<b>1. Introducción .....</b>	<b>1</b>
2. ¿Qué es Flask y para qué se utiliza? .....	2
3. Funciones principales de la librería Flask .....	3
<b>4. Casos de uso reales de Flask .....</b>	<b>4</b>
<b>5. Conclusión .....</b>	<b>7</b>
<b>6. Bibliografía .....</b>	<b>8</b>

# Introducción

En el campo del desarrollo web moderno, los frameworks se han convertido en herramientas fundamentales que permiten agilizar la creación de sitios y aplicaciones dinámicas. Estos frameworks ofrecen estructuras predefinidas que facilitan la implementación de componentes esenciales como la gestión de rutas, el manejo de peticiones HTTP, la conexión a bases de datos y la generación de interfaces visuales.

Dentro del ecosistema de **Python**, uno de los lenguajes de programación más versátiles y utilizados actualmente, destaca **Flask**, un microframework que combina potencia, flexibilidad y simplicidad.

Flask fue creado en el año 2010 por **Armin Ronacher** como parte del proyecto **Pocoo**, y desde entonces ha ganado gran popularidad entre desarrolladores y empresas por su capacidad de crear aplicaciones web y APIs RESTful con una estructura ligera y adaptable.

Su filosofía de diseño se basa en ofrecer las herramientas esenciales para construir aplicaciones web sin imponer una arquitectura rígida, lo que le permite al programador decidir qué componentes incorporar y cómo estructurar el proyecto.

El presente trabajo tiene como propósito describir en qué consiste la librería Flask, sus principales funciones y características, y analizar casos reales en los que ha sido implementada exitosamente.

## ¿Qué es Flask y para qué se utiliza?

**Flask** es un **microframework web de código abierto** desarrollado en Python. Se le denomina “micro” porque no incluye por defecto componentes complejos como sistemas de autenticación, manejo de bases de datos o validación de formularios. En lugar de eso, Flask proporciona una base mínima sobre la cual el desarrollador puede construir libremente, añadiendo extensiones o librerías externas según las necesidades del proyecto.

Flask se utiliza principalmente para:

- **Desarrollar aplicaciones web dinámicas**, que generen contenido HTML de forma automatizada.
- **Construir APIs RESTful**, que permiten la comunicación entre servidores y clientes (por ejemplo, aplicaciones móviles o servicios externos).
- **Crear prototipos o sistemas pequeños**, por su facilidad de instalación y bajo consumo de recursos.
- **Implementar microservicios**, dentro de arquitecturas distribuidas.

## Funciones principales de la librería Flask

A continuación, se describen algunas de las funciones y componentes más utilizados en Flask:

### 3.1. Flask()

Es la clase principal del framework. Permite crear una instancia de la aplicación que actuará como el servidor web.

```
from flask import Flask  
app = Flask(__name__)
```

Con esta línea se inicializa la aplicación Flask, a partir de la cual se definirán rutas, vistas y configuraciones.

### 3.2. @app.route()

Es un **decorador** que asocia una función de Python con una URL específica. Permite definir las rutas del sitio web.

```
@app.route('/')
def home():
    return 'Bienvenido a mi aplicación Flask'
```

En este ejemplo, cuando el usuario accede a la raíz del sitio, se ejecuta la función `home()`.

### 3.3. render\_template()

Esta función se utiliza para **mostrar archivos HTML dinámicos** mediante el motor de plantillas **Jinja2**, que permite integrar variables y estructuras de control en el código HTML.

```
from flask import render_template
@app.route('/usuario/<nombre>')
def usuario(nombre):
    return render_template('usuario.html', nombre=nombre)
```

De esta manera, la página puede mostrar información personalizada según el usuario.

### 3.4. request

El objeto `request` permite acceder a los datos enviados por el cliente, como formularios, parámetros de URL o datos en formato JSON.

```
from flask import request
@app.route('/login', methods=['POST'])
def login():
    usuario = request.form['usuario']
    return f'Bienvenido, {usuario}'
```

Es esencial en la creación de formularios o servicios web que reciben información del usuario.

### 3.5. `redirect()` y `url_for()`

Estas funciones permiten **redirigir al usuario** a otras rutas dentro de la aplicación.

```
from flask import redirect, url_for
@app.route('/salir')
def salir():
    return redirect(url_for('home'))
```

Esto es útil para mantener la navegación controlada dentro de la aplicación.

## Casos de uso reales de Flask

Flask ha sido utilizado en múltiples proyectos de gran relevancia internacional debido a su flexibilidad y facilidad de integración con otros servicios:

- **Netflix:** utiliza Flask en varios de sus microservicios internos para administrar datos de usuarios y manejar partes de su infraestructura de streaming.
- **Reddit:** ha empleado Flask en herramientas internas para gestión de contenido y moderación.
- **Lyft:** la compañía de transporte utiliza Flask para construir APIs que conectan servicios de datos en tiempo real.
- **MIT (Massachusetts Institute of Technology):** usa Flask en proyectos educativos y sistemas de investigación científica.
- **Startups y proyectos personales:** es ideal para emprendedores y programadores independientes que buscan desarrollar rápidamente aplicaciones web o APIs funcionales con pocos recursos.

## **Conclusión**

La librería Flask representa una de las herramientas más eficientes y versátiles para el desarrollo web en Python. Su filosofía de simplicidad y modularidad la convierte en una excelente opción tanto para principiantes como para profesionales que buscan construir aplicaciones rápidas y personalizables.

Además, su comunidad activa y su amplia documentación facilitan el aprendizaje y la resolución de problemas.

En resumen, Flask se ha consolidado como un framework confiable para crear desde pequeños prototipos hasta sistemas complejos, demostrando su relevancia en el mundo del desarrollo web moderno.

## Bibliografía

1. Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
2. Pallets Projects (2025). *Flask Official Documentation*. Disponible en: <https://flask.palletsprojects.com/>
3. Real Python (2024). *Flask by Example: Building Web Applications with Python*. Disponible en: <https://realpython.com/flask-by-example/>
4. Python Software Foundation (2025). *Python Official Documentation*. <https://docs.python.org/>
5. GeeksforGeeks (2025). *Introduction to Flask Framework in Python*. <https://www.geeksforgeeks.org/flask-framework/>