



Gracias a



OpenWebinars certifica que

Luis Angel Vidal Miguez

Ha superado con éxito

**Curso de Flask Mini-Framework
Python**

Duración del curso

14 horas

Fecha de expedición

04 mayo 2020

A handwritten signature in black ink that reads 'Manuel Agudo'.

CEO de OpenWebinars

Manuel Agudo

Contenido

Curso de Flask Mini-Framework Python

1. INTRODUCCIÓN A LA PROGRAMACIÓN WEB

Presentación del curso y del profesor

Protocolo HTTP

Programación web dinámica

Patrón modelo-vista-controlador

Introducción a la programación web con python

2. INTRODUCCIÓN A FLASK

Introducción a flask

Instalación de flask

Corriendo una aplicación sencilla

Controlando nuestra aplicación con flask-script

3. CONTROLADOR: ENRUTAMIENTO, PETICIONES Y RESPUESTAS

Enrutamiento: rutas

Enrutamiento: Métodos

Trabajando con peticiones HTTP

Generando respuestas HTTP, respuestas de error y redirecciones

Contenido estático

4. VISTA: PLANTILLAS Y FORMULARIOS

Plantillas con jinja2

Generando páginas HTML con Flask y Jinja2

Herencia de plantillas

Plantillas con bootstrap (flask-bootstrap)

Trabajando con formularios

Generando formularios con flask-wtf

Subida de ficheros

5. MODELO: GESTIONANDO LOS DATOS

Usando base de datos en Flask, flask-sqlalchemy

El modelo de base de datos I

El modelo de base de datos II

Listando y filtrando los registros de la base de datos

Creando registros en la base de datos

Modificando registros en la base de datos

Borrando un registro de la base de datos

6. GESTIÓN DE USUARIOS

Gestión de usuarios con sesiones

Registrando nuevos usuarios

Gestión de permisos de usuarios

Gestión de usuarios con extensión Flask-Login

7. GESTIÓN CARRITO DE LA COMPRA

Uso de cookies para añadir artículos al carrito I

Uso de cookies para añadir artículos al carrito II

Finalización de la compra

8. DESPLIEGUE DE LA APLICACIÓN

Despliegue de aplicación flask en un servidor LAMP

Despliegue de aplicación flask en hosting pythonanywhere

Despliegue de aplicación flask en un PaaS Heroku

Despliegue de aplicación flask en un contenedor docker