CONTENIDO

Introducción 1

CAPÍTULO 1
Patrones y arquitecturas de *software* 3
1.1. Introducción 3
1.2. Patrones de arquitectura 9
1.2.1. Patrón de estructuración 10
1.2.2. Aplicación monolítica 10
1.2.3. Capas jerárquicas 11
1.2.4. Procesamiento batch o secuencial 14
1.2.5. Tuberías y filtros (Pipes and Filters) 15
1.2.6. Microkernel 16
1.2.7. Arquitecturas orientadas a servicios (SOA) 17
1.2.8. Arquitectura REST 18
1.2.9. Arquitecturas dirigidas por eventos 20
1.2.10. Patrón productor-consumidor 25
1.3. Patrones sobre sistemas distribuidos 27
1.3.1. Bróker («agente intermediario») 27
1.4. Patrones sobre sistemas interactivos 28
1.4.1. MVC (Modelo Vista Controlador) 28
1.4.2. MVT (Modelo Vista Template) 32
1.4.3. MVC (Modelo Vista Controlador) + Observer 34
1.4.4. Implementación MVC (Modelo Vista Controlador) en Python 35
1.5. Arquitecturas basadas en microservicios 38
1.6. Retos de las arquitecturas basadas en microservicios 41
1.7. Clean Architecture 43
1.7.1. Principios de diseño 45
1.8. Conclusiones 48

CAPÍTULO 2
Python Web *Frameworks* 53
2.1. Introducción 53
2.2. TurboGears 55
2.3. Pyramid 57
2.4. CherryPy 58
2.5. BottlePy 61
2.6. Starlette 63
2.7. Responder 67
2.8. Molten 68
2.9. Klein 69
2.10. Quart 72
2.11. BlackSheep 73

2.12. FastAPI 75
2.12.1. Tratamiento de rutas con FastAPI 81
2.12.2. Tratamiento de excepciones con FastAPI 89
2.12.3. Tratamiento con SQLite 99
2.12.4. Estructura de un proyecto 104
2.13. Emmet 116
2.14. PY4WEB 118
2.15. Falcon Framework 121
2.16. Pynecone 123
2.17. Ejecutar Python en el navegador 129

CAPÍTULO 3
Django *Framework* 131
3.1. Introducción 131
3.2. Patrón Modelo-Vista-Template en Django 132
3.3. Ventajas de Django 134
3.4. Proyectos web que usan Django 135
3.5. Instalación de Django con Virtualenv 136
3.6. Crear un proyecto Django 137
3.6.1. Cómo procesa una petición Django 142
3.7. Crear una aplicación en Django 144
3.7.1. Vistas en Django 145
3.7.2. Plantillas en Django 155
3.7.3. Modelos en Django 162
3.7.4. Panel de administración en Django 171
3.8. Django-environ 179
3.9. Django REST Framework 181
3.9.1. Componentes de Django REST Framework 184
3.9.2. Serializadores con Django REST Framework 188
3.9.3. Probando la API con Django REST Framework 194
3.9.4. Creando un serializador personalizado 202
3.10. Django REST Swagger 205
3.11. Django Ninja 207
3.12. Autenticación y autorización en Django 211
3.13. Django AdminApp 214
3.14. Django TodoApp 220
3.15. Wagtail como sistema gestor de contenidos 225
3.16. Fly.io para desplegar aplicaciones Django 229

CAPÍTULO 4
Flask *Framework* 233
4.1. Introducción 233
4.2. Gestión de paquetes en Python con Pipenv y Poetry 234
4.3. Estructura de un proyecto en Flask 249
4.3.1. Añadir un template HTML 252
4.3.2. Métodos de enrutamiento 254
4.3.3. Trabajando con peticiones HTTP 255
4.3.4. Plantillas en Flask 263
4.3.5. Subir ficheros a un servidor en Flask 265
4.4. Extensiones de Flask 269
4.5. Desarrollo de API con APIFlask 270

4.6. Aplicaciones modulares con Flask 284
4.7. Flask-Injector 286
4.8. Flask-Socket.IO 289
4.9. Flask asíncrono 295
4.10. Flask-Login 298
4.11. Flask CRUD 302
4.12. Uso de Flask con JWT 310
4.12.1. PyJWT 312
4.13. Mejores prácticas en Flask 324
4.14. Flask vs. Django 325

CAPÍTULO 5
Ejecución de microservicios en Python 327
5.1. Creación de microservicios con PyMS 327
5.2. Web Server Gateway Interface (WSGI) 330
5.3. Llamadas asíncronas 332
5.4. Greenlet 334
5.5. Gevent 335
5.6. Twisted 342
5.7. Tornado 343
5.8. Peticiones asíncronas con AIOHTTP y asyncio 349
5.8.1. El módulo AIOHTTP 350
5.8.2. El módulo asyncio 352
5.8.3. Caso de uso asyncio junto con Socket.IO 377
5.9. Sanic 381
5.10. Colas de mensajes 384
5.11. RabbitMQ 385
5.11.1. Protocolo MQTT (Message Queuing Telemetry Transport) 388
5.11.2. Comunicación con el bróker 389
5.11.3. Nameko 393
5.11.4. Protocolo de comunicaciones con Nameko 397
5.11.5. Interacción con Python mediante los módulos Pika y Puka 402
5.11.6. Administración de RabbitMQ 414
5.11.7. Interacción con Python mediante AMQPStorm 416
5.11.8. Conclusiones RabbitMQ 427
5.12. Apache Kafka 427
5.13. Celery 431
5.13.1. Uso de Django con Celery 440
5.14. Aplicaciones distribuidas y asíncronas con PyZMQ 445

CAPÍTULO 6
Desarrollo de microservicios con arquitecturas Serverless 453
6.1. Introducción 453
6.2. AWS Lambda 455
6.2.1. Funciones Lambda con Python 458
6.3. Serverless en las aplicaciones Python WSGI 462
6.4. *Frameworks* Serverless 463
6.5. Slam 468
6.6. Zappa 468
6.7. Chalice 473
6.8. FaaS (Function as a Service) vs. Microservicios 480

Glosario 483