
SYLLABUS - GUÍA MLOps v5.0: SENIOR EDITION

Ruta de Aprendizaje: De Cero a Lead/Senior MLOps Engineer

DuqueOM | Noviembre 2025

SYLLABUS v5.0

Guía MLOps: Senior Edition

De Python básico a Lead/Senior MLOps Engineer

Duración Total	Nivel Inicial	Nivel Final
16-20 semanas	Python Básico	Lead/Senior MLOps

Tabla de Contenidos

1. Descripción del Curso
2. Nueva Estructura: 7 Fases
3. Prerrequisitos
4. Calendario Sugerido
5. Sistema de Evaluación
6. Recursos por Módulo

1. Descripción del Curso

1.1 ¿Qué Hace Diferente a v5.0?

EVOLUCIÓN: v4.0 → v5.0 SENIOR EDITION	
v4.0 (Practitioner)	v5.0 (Senior Edition)
<ul style="list-style-type: none">• "Usa DVC"• Código funcional• Deploy básico• Instrucciones lineales• Sin Python estructurado	<ul style="list-style-type: none">• "¿Cuándo DVC vs Git LFS vs Delta Lake?"• Código tipado + testeado + documentado• Secrets + Escalado + FinOps• Trade-offs + ADRs en cada decisión• NUEVO: Módulo Python Moderno

1.2 Perfil del Egresado

Al completar el curso, serás capaz de:

- **Diseñar** arquitecturas ML con trade-offs documentados (ADRs)
- **Escribir** código Python profesional (tipado, Pydantic, OOP)
- **Implementar** pipelines CI/CD robustos
- **Desplegar** modelos con decisiones informadas (Lambda vs ECS vs K8s)
- **Operar** sistemas en producción con observabilidad completa

- **Documentar** proyectos con Model Cards y ética
- **Defender** decisiones técnicas en entrevistas de nivel Senior

2. Nueva Estructura: 7 Fases

2.1 Mapa del Curso v5.0



2.2 Detalle por Módulo

#	Módulo	Duración	Tipo	Enfoque Senior
00	Índice y Roadmap	1h	Orientación	Visión de negocio, ROI
01	Python Moderno (NUEVO)	5h	Fundamental	Typing, Pydantic, OOP, src/ layout
02	Diseño de Sistemas	4h	Arquitectura	ML Canvas, C4 Model, ADRs
03	Entornos Profesionales	3h	Práctico	venv vs Conda vs Poetry vs Docker
04	Git Profesional	4h	Práctico	Conventional Commits, pre-commit
05	Ingeniería de Datos	4h	Práctico	DVC avanzado, cuándo NO usar DVC
06	Pipelines Avanzados	5h	ML	Custom Transformers, Data Leakage
07	Experiment Tracking	4h	ML	MLflow pro, Model Registry
08	Testing ML	5h	Calidad	Pirámide de testing para ML
09	GitHub Actions	4h	CI/CD	Caching, Matrix, Secrets
10	Docker Avanzado	5h	DevOps	Multi-stage, Distroless, CVE scan
11	FastAPI Pro	5h	API	DI, Middleware, Error Handling
12	Despliegue Híbrido	4h	Decisiones	Lambda vs ECS vs K8s + FinOps
13	Terraform Modular	5h	IaC	Módulos, Workspaces, State
14	Kubernetes ML	6h	Orquestación	HPA, Secrets, Probes
15	Observabilidad	5h	Producción	Logs, Metrics, Drift Detection
16	Documentación y Ética	4h	Profesional	MkDocs, Model Cards, Responsible AI
17	Proyecto Integrador	4h	Capstone	Demo, Pitch, Entrevistas

Total estimado: 77-85 horas

3. Prerrequisitos

3.1 Conocimientos Requeridos

Área	Nivel Mínimo	Módulo que Cubre
Python	Sintaxis, funciones, clases básicas	01 profundiza
Terminal	Navegación, comandos básicos	03, 04
Git	Conceptos básicos (opcional)	04 cubre desde cero
ML Básico	Regresión/Clasificación supervisada	06 asume conocimiento

3.2 Tiempo Disponible

Dedicación Semanal	Duración Estimada	Ritmo
5-10 horas	20 semanas	Relajado
10-15 horas	16 semanas	Normal
15-20 horas	12 semanas	Intensivo

4. Calendario Sugerido

Plan de 16 Semanas (Ritmo Normal)

Semana	Fase	Módulos	Entregables
1	1	00, 01	Código tipado + estructura src/
2	1	02	ML Canvas + ADR documentado
3	2	03	Entorno reproducible configurado
4	2	04	Git con Conventional Commits + pre-commit
5	2	05	Dataset versionado con DVC
6	3	06	Pipeline sklearn sin data leakage
7	3	07	Experimentos en MLflow + Model Registry
8	3	08	Tests pasando > 80% coverage
9	4	09	CI/CD ejecutando en GitHub Actions
10	4	10	Imagen Docker < 500MB, Trivy sin CRITICAL
11	5	11	API FastAPI con validación Pydantic
12	5	12	Decisión de despliegue documentada
13	6	13	Terraform modular funcionando
14	6	14	Deployment en K8s con HPA
15	6	15	Dashboard Grafana + alertas
16	7	16, 17	Model Card + Video Demo + Portafolio publicado

5. Sistema de Evaluación

5.1 Componentes de Evaluación

Componente	Peso	Descripción
Ejercicios	25%	Completar ejercicios de cada módulo
Código	30%	Tipado, tests, estructura profesional
Proyecto Integrador	25%	Portafolio completo funcional
Demo/Documentación	20%	Video 3-5 min + Model Card + README

5.2 Criterios de Nivel Senior

CRITERIOS DE NIVEL SENIOR
CÓDIGO:
[] Typing completo (mypy pasa)
[] Pydantic para configuración
[] Tests > 80% coverage
[] Estructura src/ layout
ARQUITECTURA:
[] ADRs documentados
[] Trade-offs explicados
[] Decisiones de despliegue justificadas
PRODUCCIÓN:
[] CI/CD completo
[] Docker optimizado
[] Observabilidad configurada
[] Drift detection implementado
PROFESIONAL:
[] Model Card completo
[] Demo profesional
[] Puede defender decisiones en entrevista

6. Recursos por Módulo

6.1 Lecturas por Fase

Fase	Recursos Clave
1	PEP 484 (Typing), Pydantic Docs, ML Canvas Paper
2	Conventional Commits, DVC Docs, Git Pro Book
3	MLflow Docs, pytest Docs, Great Expectations
4	GitHub Actions Docs, Docker Best Practices, Trivy
5	FastAPI Tutorial, AWS Lambda vs ECS comparison
6	Terraform Registry, K8s Docs, Prometheus/Grafana
7	Model Cards Papers, Responsible AI Guidelines

Navegación

Ir a...

Índice General

► Comenzar: Módulo 01 Python Moderno

Ejercicios

Glosario

Versión: 5.0 Senior Edition | **Última actualización:** Noviembre 2025

© 2025 DuqueOM - Guía MLOps: Senior Edition