MALDIVAS TECH · CURSO LIVRE · MÓDULO 0 GRATUITO (11H)

Trilha Fundamentos (F01 a F06): do zero à primeira API

Curso: Construa seu marketplace: do código ao pagamento em produção · Analogia única: o restaurante

F01 — O mapa: cada tecnologia como um posto do restaurante

Posto no restaurante	Tecnologia real	No MySuperStore
Salão e cardápio	Front-end	Next.js + React (Vercel)
Comanda	Requisição HTTP + JSON	POST /api/v1/orders/
Porteiro	Proxy reverso e HTTPS	Caddy
Maître / Garçom	Rotas (urls.py)	config/urls.py
Conferente	Serializer (DRF)	apps/orders/serializers.py
Cozinheiro	View / ViewSet	apps/orders/views.py
Despensa + Tradutor	PostgreSQL + ORM + Migração	models.py + migrations/
Auxiliar + Quadro + Interfone	Celery + Redis + WebSocket	Celery + Channels/Daphne

Onde a analogia quebra: na internet, duas pessoas clicam no último item no mesmo milissegundo e o provedor pode reenviar o mesmo webhook duas vezes. Por isso existem travas, transações e idempotência.

F02 — Preparando a bancada no Windows

Testado em Windows 11, Python 3.12.10, Django 5.1.4 e DRF 3.15.2 (13/11/2026). Sempre marque `Add python.exe to PATH` e use ambiente virtual isolado:

```
py -3.12 -m venv .venv
.venv\Scripts\python -m pip install django==5.1.4 djangorestframework==3.15.2
.venv\Scripts\python -m pip freeze > requirements.txt
```

F03 — Seu primeiro projeto Django

- **Projeto** = restaurante inteiro (`settings.py` , `urls.py`). **App** = departamento especializado (`catalogo`).
- **Model → Tabela:** sempre em dois passos: `makemigrations` (gera a planta) e `migrate` (executa a obra).
- Dinheiro vai sempre em `DecimalField` (nunca `FloatField`), com `CheckConstraint` no banco.

F04 — Sua primeira API com DRF

- **200 OK** (consulta), **201 Created** (criou), **400 Bad Request** (dado inválido), **404 Not Found** (não existe), **405 Method Not Allowed** (verbo não aceito), **409 Conflict** (ex.: estoque insuficiente).
- Campos controlados pelo servidor (como `status` e `total`) ficam em `read_only_fields` .

F05 — Testes automáticos, Git e regra dos segredos

- Padrão **Arrange-Act-Assert** rodando em banco temporário com `manage.py test` .

- Crie `.gitignore` (`.venv/`, `.env`, `db.sqlite3`) **antes** do primeiro commit. Segredo no histórico do Git deve ser considerado vazado e trocado.

F06 — Método de leitura "da porta para dentro"

Para não se perder em projetos grandes: **1.** Planta (README) → **2.** Como roda (Compose/requirements) → **3.** Departamentos (`LOCAL_APPS`) → **4.** Cardápio (`urls.py`) → **5.** Siga um pedido de ponta a ponta (rota → view → serializer → model → teste).
