

Parte 4 — VSCode, Fluxos & Anexos

```
# Parte 4 – VSCode Continue, Fluxos, Checklist, Troubleshooting e Anexos

## 13. Integração Ollama + VSCode (Continue)
Exemplo `.continue/config.json`:
```json
{
 "version": "2.0.0",
 "models": [
 {
 "name": "Assistente Principal", "provider": "ollama", "model": "llama3.1:8
b", "roles": ["chat", "edit"], "defaultCompletionOptions": { "temperature": 0.6, "
maxTokens": 4096 } },
 {
 "name": "Especialista em
Código", "provider": "ollama", "model": "deepseek-coder-
v2:16b", "roles": ["edit", "review"], "defaultCompletionOptions": { "temperature
": 0.2, "maxTokens": 8192 } }
],
 "context":
[{ "provider": "codebase", "params": { "nRetrieve": 30, "nFinal": 10 } },
{ "provider": "file" }, { "provider": "terminal" }],
 "prompts": [{ "name": "revisao-completa", "description": "Revisão detalhada
de código", "prompt": "Analise o código selecionado considerando: qualidade,
segurança, performance e sugira melhorias." }]
}
```

## 14. Fluxos de trabalho recomendados
**Projeto FastAPI + Ollama**
1. Criar venv e instalar dependências (fastapi, uvicorn)
2. Baixar modelo (deepseek-coder-v2)
3. Usar Continue para gerar docs/testes
4. Testar local → staging → produção

**Ciência de dados**
- Use `conda` para dependências compiladas
- Salve `environment.yml`, seeds, resultados e scripts de preprocessing

## 15. Checklist de produção
**Ambientes**
- [ ] Isolar ambiente por projeto (.venv / conda)
- [ ] Versionar `requirements.txt` ou `environment.yml`

**Modelos/IA**
- [ ] Registrar versão do modelo e parâmetros (temperature, max_tokens)
- [ ] Revisão humana para outputs críticos

**Segurança/Código**
- [ ] Scanne vulnerabilidades (Dependabot, safety)
- [ ] Não versionar secrets (use secrets manager)

## 16. Resolução de problemas (troubleshooting)
**PowerShell não ativa**
```powershell
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
& ".venv\Scripts\Activate.ps1"
```

**`ollama`: modelo não responde
```

```

```bash
ollama list
ollama logs
ollama rm <model>
ollama pull <model>
```

**Pacotes faltando após ativar venv**
- Verifique `which python` / `where python`
- Recrie ambiente: `python -m venv --clear .venv`

## 17. Anexos: snippets e templates
**requirements.txt exemplo**
```
fastapi==0.95.0
uvicorn[standard]==0.22.0
pydantic==1.10.7
requests==2.31.0
```

**environment.yml exemplo**
```yaml
name: exemplo_proj
channels:
 - conda-forge
dependencies:
 - python=3.10
 - numpy=1.23
 - pandas=1.5
 - pip:
 - fastapi==0.95.0
```

**Script de setup (exemplo completo)**
```bash
#!/usr/bin/env bash
Uso: ./setup.sh <diretorio_projeto>
set -e
DIR=${1:-.}
python3 -m venv $DIR/.venv
source $DIR/.venv/bin/activate
pip install --upgrade pip
if [-f $DIR/requirements.txt]; then
 pip install -r $DIR/requirements.txt
fi
```

```