

# Guia de Execução Passo a Passo

## Execução do Setup Completo

### 1. Preparação do Ambiente

```
bash
```

```
# 1. Clone ou acesse seu projeto
```

```
cd seu-projeto
```

```
# 2. Certifique-se que tem Python 3.8+ e Git
```

```
python3 --version
```

```
git --version
```

```
# 3. Se não for um repositório Git, inicialize
```

```
git init
```

### 2. Criação dos Arquivos de Setup

Crie cada um dos arquivos principais em sequência:

#### a) Estrutura Base

```
bash
```

```
# Criar estrutura inicial
```

```
mkdir -p .framework/config
```

```
mkdir -p .claude/context
```

```
mkdir -p scripts/{validation,automation}
```

```
mkdir -p docs/prd
```

#### b) Arquivo 1: Configuração das Etapas

```
bash
```

```
# Copie o conteúdo do "Configuração do TaskMaster AI" para:
```

```
nano .framework/config/stages.yaml
```

#### c) Arquivo 2: Mapeamento PRD

```
bash
```

```
# Copie o conteúdo do "Mapeamento PRD para Framework" para:
```

```
nano .framework/config/prd-mapping.yaml
```

### d) Arquivo 3: Script de Validação

bash

*# Copie o conteúdo do "Scripts de Validação PRD-Framework" para:*

`nano scripts/validation/prd_validator.py`

`chmod +x scripts/validation/prd_validator.py`

### e) Arquivo 4: Context Loader do Claude

bash

*# Copie o conteúdo do "Prompts Específicos por Etapa" para:*

`nano .claude/context/context_loader.py`

### f) Arquivo 5: Template Base do Claude

bash

*# Copie o conteúdo do "Integração Claude Code com Framework" para:*

`nano .claude/context/base-context.md`

### g) Arquivo 6: Script de Automação

bash

*# Copie o conteúdo do "Scripts de Automação e Integração" para:*

`nano scripts/automation/framework-automation.sh`

`chmod +x scripts/automation/framework-automation.sh`

### h) Arquivo 7: Setup de Git Hooks

bash

*# Copie o conteúdo do "Git Hooks para Validação Automática" para:*

`nano scripts/automation/setup-git-hooks.sh`

`chmod +x scripts/automation/setup-git-hooks.sh`

### i) Arquivo 8: Setup Completo

bash

*# Copie o conteúdo do "Setup Completo - Comando de Inicialização" para:*

`nano setup-framework.sh`

`chmod +x setup-framework.sh`

### 3. Execução do Setup

```
bash

# Executar setup completo
./setup-framework.sh
```

### 4. Verificação da Instalação

```
bash

# Verificar status
./scripts/automation/framework-automation.sh status

# Testar validação
./scripts/automation/framework-automation.sh validate

# Verificar contexto do Claude
python3 .claude/context/context_loader.py context
```

## Configuração Pós-Instalação

#### 1. Configurar TaskMaster AI

Se você estiver usando TaskMaster AI, configure-o para usar os arquivos do framework:

```
bash

# No seu TaskMaster AI, configure para:
# - Ler .framework/config/stages.yaml
# - Monitorar .framework/current_stage.json
# - Executar validações via scripts/validation/prd_validator.py
```

#### 2. Configurar Claude Code

No VS Code com Claude Code:

```
json

// Adicione às configurações do VS Code
{
  "claude.contextFiles": [
    ".claude/context/current-context.md",
    "docs/prd/main-prd.md",
    "docs/framework/progress-tracker.md"
  ]
}
```

### 3. Configurar PRD Inicial

```
bash

# Editar PRD principal
nano docs/prd/main-prd.md

# Definir personas
nano docs/prd/personas.md

# Definir requisitos técnicos
nano docs/prd/technical-requirements.md
```

### Workflow Diário

#### Manhã: Check de Status

```
bash

# Verificar onde estamos
./scripts/automation/framework-automation.sh status

# Atualizar contexto do Claude
./scripts/automation/framework-automation.sh update-context
```

#### Durante o Desenvolvimento

```
bash

# Ao fazer commit (automático via git hooks)
git add .
git commit -m "feat: implementar validação de usuário"
# -> Validação automática será executada

# Para perguntas ao Claude Code
python3 .claude/context/context_loader.py prompt "Como implementar autenticação JWT para Stage 2?"
```

#### Final do Dia: Validação e Progresso

bash

*# Validação completa*

`./scripts/automation/framework-automation.sh validate`

*# Gerar relatório*

`./scripts/automation/framework-automation.sh report`

*# Se tudo OK, considerar avanço*

`./scripts/automation/framework-automation.sh advance`

## Troubleshooting Comum

### Erro: "Python3 não encontrado"

bash

*# Instalar Python (Ubuntu/Debian)*

`sudo apt update && sudo apt install python3 python3-pip`

*# Instalar Python (macOS)*

`brew install python3`

*# Instalar Python (Windows)*

*# Baixe de [python.org](https://python.org)*

### Erro: "Arquivo YAML inválido"

bash

*# Verificar sintaxe YAML*

`python3 -c "import yaml; yaml.safe_load(open('.framework/config/stages.yaml'))"`

### Erro: "Git hooks não funcionam"

bash

*# Reconfigurar hooks*

`./scripts/automation/setup-git-hooks.sh`

*# Verificar permissões*

`ls -la .git/hooks/`

### Contexto do Claude não atualiza

bash

*# Forçar atualização*

`python3 .claude/context/context_loader.py generate`








*# Verificar se arquivo foi criado*

`cat .claude/context/current-context.md`



## Métricas de Sucesso

Após o setup, você deve ter:

-  8 etapas configuradas no framework
-  Validação automática funcionando
-  Contexto do Claude atualizado
-  Git hooks operacionais
-  Dashboard de progresso ativo
-  PRD template preenchido
-  Scripts de automação funcionais



## Próximos Passos

1. **Completar PRD:** Documentar problema do cliente
2. **Começar Stage 1:** Executar customer discovery
3. **Usar Validação:** Validar progresso continuamente
4. **Iterar:** Usar feedback para melhorar



## Dicas de Uso

### TaskMaster AI Integration

- Use `taskmaster validate-prd --stage=current` para validações
- Configure tasks automáticas baseadas nas etapas
- Integre com seu workflow de planning

### Claude Code Optimization

- O contexto é atualizado automaticamente a cada commit
- Use prompts específicos por etapa para melhor orientação
- Contexto inclui sempre status atual do PRD e framework

### Colaboração em Equipe

- Cada membro da equipe deve executar o setup
- Git hooks garantem validação consistente
- Dashboard compartilha progresso com todos

Agora você tem um sistema completo integrado! 🚀