

Casos de Uso - Orange Pi Provisioning System

Este documento detalha casos de uso específicos e cenários reais de aplicação do sistema de provisionamento.

Índice

- [Casos de Uso Principais](#)
- [Cenários de Produção](#)
- [Casos de Uso Avançados](#)
- [Integração com Outros Sistemas](#)
- [Casos de Uso Educacionais](#)

Casos de Uso Principais

1. Impressora 3D Ender 3 SE com Klipper

Contexto: Automatizar setup de Orange Pi Zero 3 para controle de impressora 3D.

Objetivo: Configurar sistema completo com Klipper, interface web e monitoramento.

Pré-requisitos:

- Orange Pi Zero 3 (2GB RAM)
- MicroSD 16GB+ (Classe 10)
- Impressora Ender 3 SE
- Cabo USB para conexão com impressora
- Rede WiFi estável

Fluxo de Execução:

```
# 1. Preparar ambiente
docker compose build

# 2. Coletar informações do sistema
docker compose run --rm provisioner scripts/collect-local-info.sh

# 3. Executar deploy específico
docker compose run --rm --privileged provisioner scripts/deploy-ender3.sh

# 4. Inserir MicroSD no Orange Pi
# Conectar cabo USB à impressora
# Aguardar inicialização (5-10 minutos)

# 5. Validar instalação
docker compose run --rm provisioner scripts/validate-deployment.sh ender3
```

Resultado Esperado:

- Sistema Klipper funcionando
- Interface web acessível em <http://192.168.1.100>

- Conexão USB com impressora estabelecida
- Configuração de firmware aplicada
- Tela LCD funcionando (se aplicável)

Validações Automáticas:

- ☒ Ping para 192.168.1.100
- ☒ SSH funcionando (porta 22)
- ☒ Serviço Klipper ativo
- ☒ Interface web respondendo (porta 80)
- ☒ Conexão USB com impressora
- ☒ Configuração de firmware carregada

2. Máquina de Corte a Laser LaserTree K1

Contexto: Configurar Orange Pi Zero 2W para controle de máquina de corte a laser.

Objetivo: Sistema completo com LightBurn Bridge e controle de laser.

Pré-requisitos:

- Orange Pi Zero 2W (1GB RAM)
- MicroSD 8GB+ (Classe 10)
- Máquina LaserTree K1
- Cabo de controle (USB/Serial)
- Rede WiFi estável

Fluxo de Execução:

```
# 1. Executar deploy específico
docker compose run --rm --privileged provisioner scripts/deploy-laser.sh

# 2. Configurar hardware
# Inserir MicroSD no Orange Pi
# Conectar cabo de controle à máquina
# Aguardar inicialização

# 3. Validar instalação
docker compose run --rm provisioner scripts/validate-deployment.sh laser
```

Resultado Esperado:

- LightBurn Bridge funcionando
- Interface de controle acessível em <http://192.168.1.101>
- Comunicação com máquina estabelecida
- Configurações de segurança ativas
- Sistema de emergência configurado



Cenários de Produção

1. Laboratório de Fabricação Digital (FabLab)

Cenário: FabLab com múltiplas impressoras 3D e máquinas de corte.

Configuração:

```
{
  "lab_config": {
    "impressoras": [
      {"id": "ender3-01", "ip": "192.168.1.100", "tipo": "ender3"},
      {"id": "ender3-02", "ip": "192.168.1.102", "tipo": "ender3"},
      {"id": "ender3-03", "ip": "192.168.1.103", "tipo": "ender3"}
    ],
    "lasers": [
      {"id": "laser-01", "ip": "192.168.1.101", "tipo": "laser"},
      {"id": "laser-02", "ip": "192.168.1.104", "tipo": "laser"}
    ]
  }
}
```

Processo de Deploy em Massa:

```
#!/bin/bash
# Script para deploy em massa

DEVICES=("ender3" "ender3" "ender3" "laser" "laser")
IPS=("192.168.1.100" "192.168.1.102" "192.168.1.103" "192.168.1.101" "192.168.1.104")

for i in "${!DEVICES[@]}; do
  echo "Configurando dispositivo ${DEVICES[$i]} com IP ${IPS[$i]}"

  # Modificar configuração temporariamente
  jq ".projects.${DEVICES[$i]}.network.static_ip = \"${IPS[$i]}\"" \
    configs/projects-config.json > temp_config.json
  mv temp_config.json configs/projects-config.json

  # Executar deploy
  docker compose run --rm --privileged provisioner scripts/deploy-${DEVICES[$i]}.sh

  echo "Aguarde inserir próximo MicroSD..."
  read -p "Pressione Enter para continuar..."
done
```

2. Ambiente de Produção Industrial

Cenário: Linha de produção com controle automatizado.

Características:

- Monitoramento 24/7
- Backup automático de configurações
- Alertas por email/SMS
- Integração com sistemas ERP

Configuração de Monitoramento:

```
# Adicionar ao crontab do sistema host
# Monitoramento a cada 5 minutos
*/5 * * * * docker compose run --rm provisioner scripts/validate-deployment.sh ender3
--silent

# Backup diário das configurações
0 2 * * * docker compose run --rm provisioner tar -czf /backup/configs-$(date +
%Y%m%d).tar.gz configs/ state/
```

3. Ambiente Educacional

Cenário: Escola técnica com aulas práticas de fabricação digital.

Configuração para Sala de Aula:

```
# Script para reset rápido entre aulas
#!/bin/bash
reset_classroom() {
    echo "🔄 Resetando ambiente de sala de aula..."

    # Limpar estados anteriores
    rm -f state/*-deployment.json

    # Restaurar configurações padrão
    cp configs/classroom-defaults.json configs/projects-config.json

    echo "✅ Ambiente pronto para nova aula"
}
```



Casos de Uso Avançados

1. Deploy Remoto via SSH

Cenário: Configurar Orange Pi remotamente sem acesso físico.

```
# Configurar deploy remoto
export REMOTE_HOST="192.168.1.50"
export REMOTE_USER="admin"

# Executar deploy via SSH
ssh $REMOTE_USER@$REMOTE_HOST "
    cd orange-pi-provisioning
    docker compose run --rm --privileged provisioner scripts/deploy-ender3.sh
"
```

2. Integração com CI/CD Pipeline

Cenário: Automatizar deploy como parte de pipeline de desenvolvimento.

```
# .github/workflows/deploy.yml
name: Deploy to Orange Pi

on:
  push:
    tags:
      - 'v*'

jobs:
  deploy:
    runs-on: self-hosted
    steps:
      - uses: actions/checkout@v4

      - name: Deploy to Production
        run: |
          docker compose build
          docker compose run --rm --privileged provisioner scripts/deploy-ender3.sh
          docker compose run --rm provisioner scripts/validate-deployment.sh ender3
```

3. Configuração Multi-Tenant

Cenário: Múltiplos clientes com configurações isoladas.

```
# Estrutura para multi-tenant
mkdir -p tenants/{cliente1,cliente2,cliente3}

# Configuração por cliente
cp configs/projects-config.json tenants/cliente1/
cp configs/projects-config.json tenants/cliente2/

# Deploy específico por cliente
TENANT=cliente1 docker compose run --rm --privileged \
  -v $(pwd)/tenants/$TENANT:/app/configs \
  provisioner scripts/deploy-ender3.sh
```

Integração com Outros Sistemas

1. Integração com OctoPrint

Configuração:

```
# Adicionar OctoPrint ao deploy Ender3
# Modificar scripts/deploy-ender3.sh

install_octoprint() {
    echo "📦 Instalando OctoPrint..."

    # Instalar dependências
    apt-get update
    apt-get install -y python3-pip python3-dev

    # Instalar OctoPrint
    pip3 install OctoPrint

    # Configurar serviço
    systemctl enable octoprint
    systemctl start octoprint
}
```

2. Integração com Home Assistant

Configuração MQTT:

```
# Adicionar ao deploy
mqtt:
    broker: "192.168.1.10"
    port: 1883
    topics:
        status: "homeassistant/3dprinter/ender3/status"
        temperature: "homeassistant/3dprinter/ender3/temperature"
        progress: "homeassistant/3dprinter/ender3/progress"
```

3. Integração com Prometheus/Grafana

Métricas de Monitoramento:

```
# Adicionar exporters de métricas
install_monitoring() {
    # Node Exporter para métricas do sistema
    wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/
node_exporter-1.6.1.linux-armv7.tar.gz
    tar xvfz node_exporter-1.6.1.linux-armv7.tar.gz
    cp node_exporter-1.6.1.linux-armv7/node_exporter /usr/local/bin/

    # Configurar serviço
    systemctl enable node_exporter
    systemctl start node_exporter
}
```

Casos de Uso Educacionais

1. Curso de IoT e Automação

Módulo 1: Configuração básica de Orange Pi

```
# Exercício prático
docker compose run --rm provisioner scripts/collect-local-info.sh
# Alunos analisam dados coletados
```

Módulo 2: Deploy automatizado

```
# Exercício com modo dry-run
export DRY_RUN=true
docker compose run --rm provisioner scripts/deploy-ender3.sh
```

Módulo 3: Validação e troubleshooting

```
# Exercício de diagnóstico
docker compose run --rm provisioner scripts/validate-deployment.sh ender3
```

2. Workshop de DevOps

Atividade 1: Containerização

- Análise do Dockerfile
- Modificação de configurações
- Build personalizado

Atividade 2: CI/CD Pipeline

- Configuração de GitHub Actions
- Testes automatizados
- Deploy automatizado

Atividade 3: Monitoramento

- Configuração de logs
- Métricas de performance
- Alertas automatizados

3. Projeto Final - Sistema Personalizado

Objetivo: Criar configuração personalizada para projeto específico.

Entregáveis:

1. Configuração JSON personalizada
2. Script de deploy customizado
3. Documentação completa
4. Testes de validação
5. Apresentação do projeto

Template de Projeto:

```
{
  "student_project": {
    "name": "Projeto Personalizado",
    "description": "Descrição do projeto",
    "hardware": {
      "board": "Orange Pi Zero 3",
      "peripherals": ["Camera", "Sensors", "Actuators"]
    },
    "software": {
      "base_os": "Armbian",
      "applications": ["Custom App 1", "Custom App 2"],
      "services": ["Web Server", "Database", "API"]
    },
    "network": {
      "static_ip": "192.168.1.200",
      "hostname": "student-project-pi",
      "ports": [80, 443, 8080]
    }
  }
}
```

Métricas e KPIs

Métricas de Sucesso por Caso de Uso

Caso de Uso	Tempo Deploy	Taxa Sucesso	Tempo Validação
Ender3 SE	< 15 min	> 95%	< 5 min
Laser K1	< 12 min	> 90%	< 3 min
Deploy Massa	< 10 min/unit	> 85%	< 2 min/unit
Educacional	< 20 min	> 80%	< 10 min

Indicadores de Performance

- **MTTR** (Mean Time To Recovery): < 30 minutos
- **MTBF** (Mean Time Between Failures): > 720 horas
- **Uptime**: > 99.5%
- **User Satisfaction**: > 4.5/5.0

Casos de Uso Futuros

1. Suporte a Novos Hardwares

- Orange Pi 5 (8GB)
- Raspberry Pi 4/5
- Rock Pi 4
- Banana Pi

2. Novos Projetos

- CNC Router

- Plotter de Vinil
- Estação Meteorológica
- Sistema de Segurança

3. Funcionalidades Avançadas

- Deploy via API REST
- Interface web para configuração
- Mobile app para monitoramento
- Integração com cloud services

Nota: Este documento é atualizado regularmente conforme novos casos de uso são identificados e implementados.