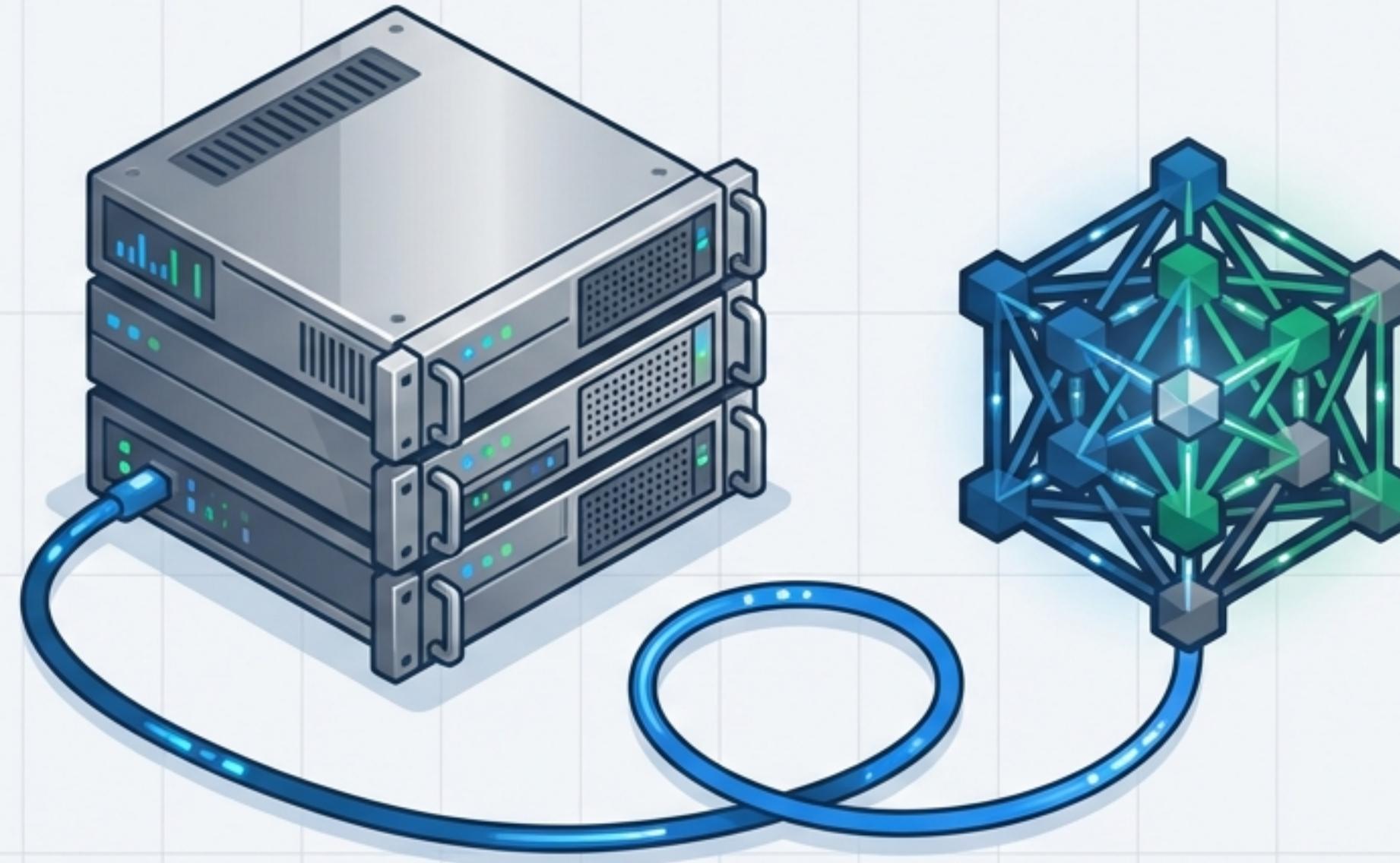


Sistema Modular de Automação

Gerenciamento inteligente de servidores Linux, atualizações e containers Docker.



100% desenvolvido por IA ([GitHub Copilot](#)) sob supervisão humana.

O Que é o Ecossistema de Automação?

Um conjunto de ferramentas modulares projetado para orquestrar a configuração, manutenção e atualização de servidores Ubuntu e ambientes Docker.



Reutilização

Componentes divididos em bibliotecas independentes em `lib/`.



Confiabilidade

Validado por mais de 100 testes unitários automatizados.



Simplicidade

Instalação via bootstrap único e operações sem intervenção manual.

Tech Stack:



Zsh>_

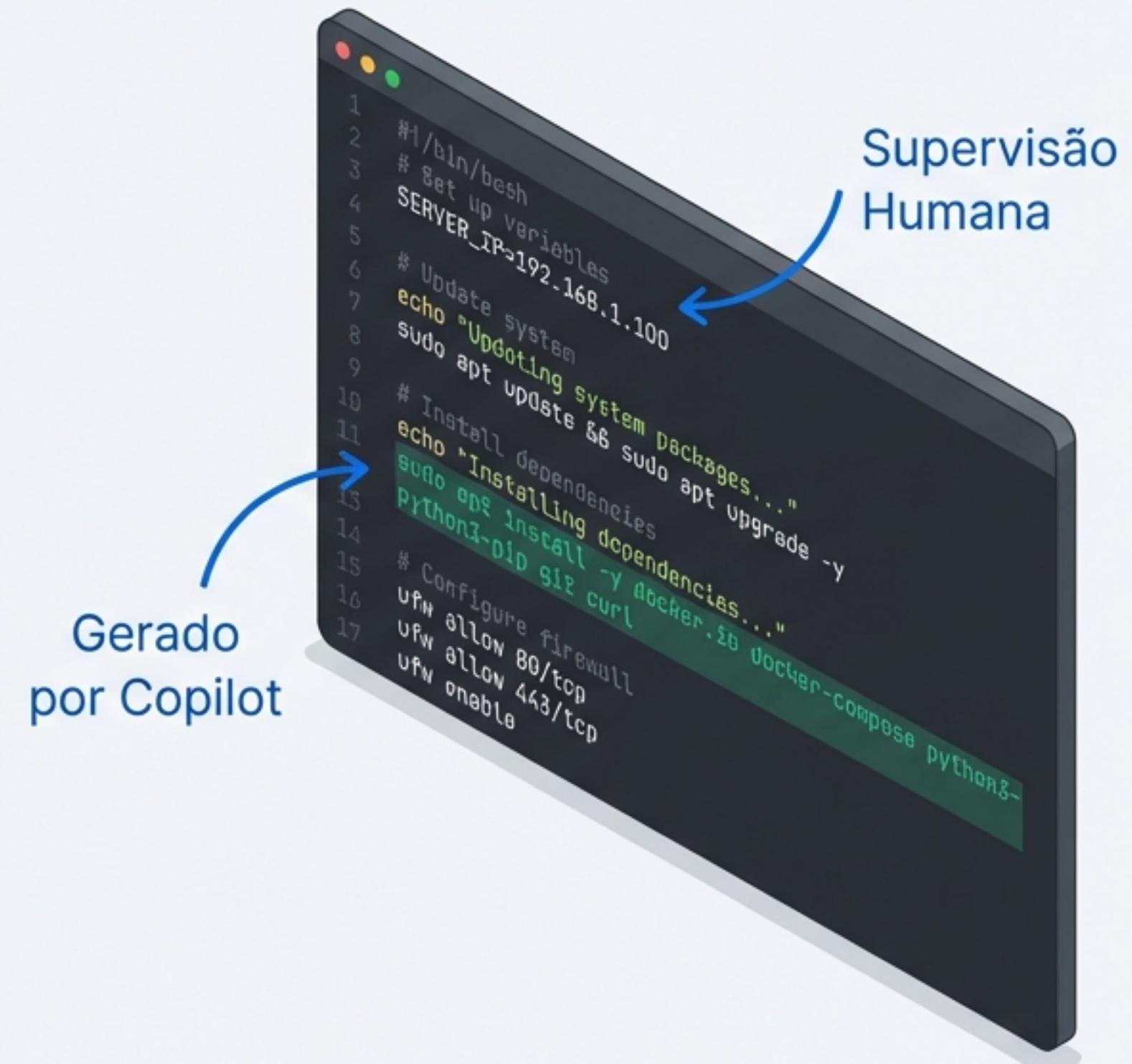


Engenharia Assistida por IA

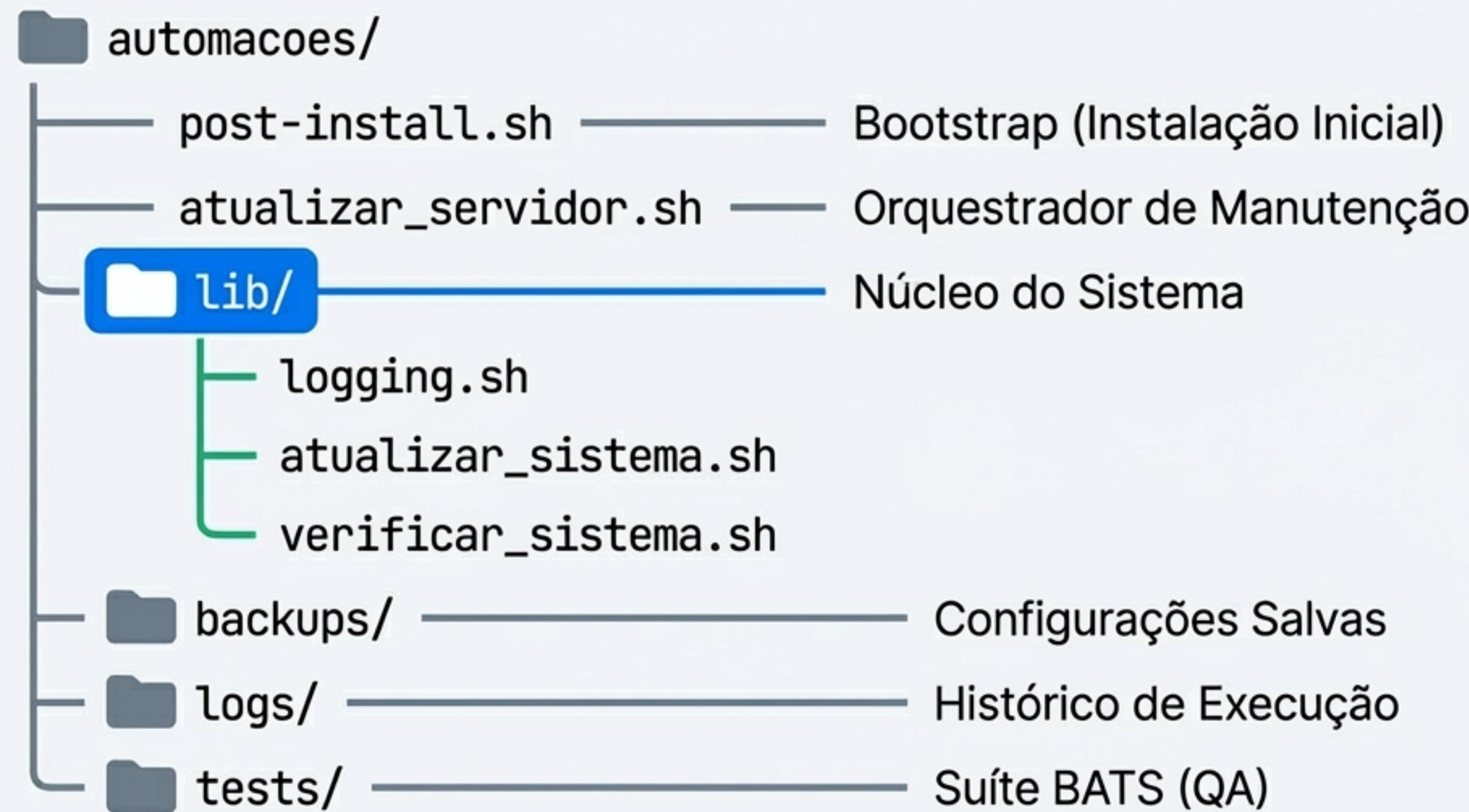
Metodologia de Desenvolvimento

Todo o código fonte, documentação e testes foram gerados utilizando GitHub Copilot (modelo Claude Sonnet 4.5), sob a supervisão técnica do analista Marcelo Ratton.

-  **Código Limpo:** Estruturas padronizadas e amplamente comentadas.
-  **Prevenção de Falhas:** Mecanismos proativos de análise de erros.
-  **Documentação Completa:** READMEs detalhados e Changelogs em tempo real.



Arquitetura do Repositório



A estrutura modular permite que cada script funcione de forma isolada ou em conjunto.

Fase 1: Instalação e Bootstrap



```
user@server:~$  
wget https://raw.githubusercontent.com/rattones/automacoes/main/post-install.sh  
chmod +x post-install.sh  
../post-install.sh
```

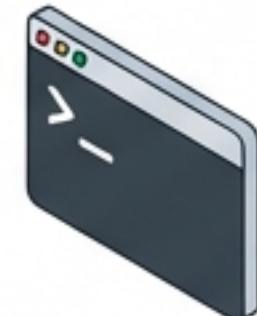
O script instala dependências mínimas (git, curl), clona o repositório e inicia o orquestrador `lib/post-install/main-install.sh`.

O Stack: O Que Será Instalado?

System Core



OpenSSH Server: Acesso remoto seguro.

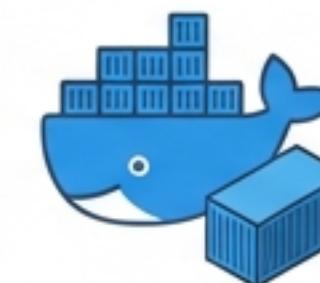


Zsh + Powerlevel10k: Terminal moderno com temas e plugins.



Cockpit Web Console: Interface GUI na porta 9090.

Developer Tools



Docker + Compose: Engine de containers (setup sem root).

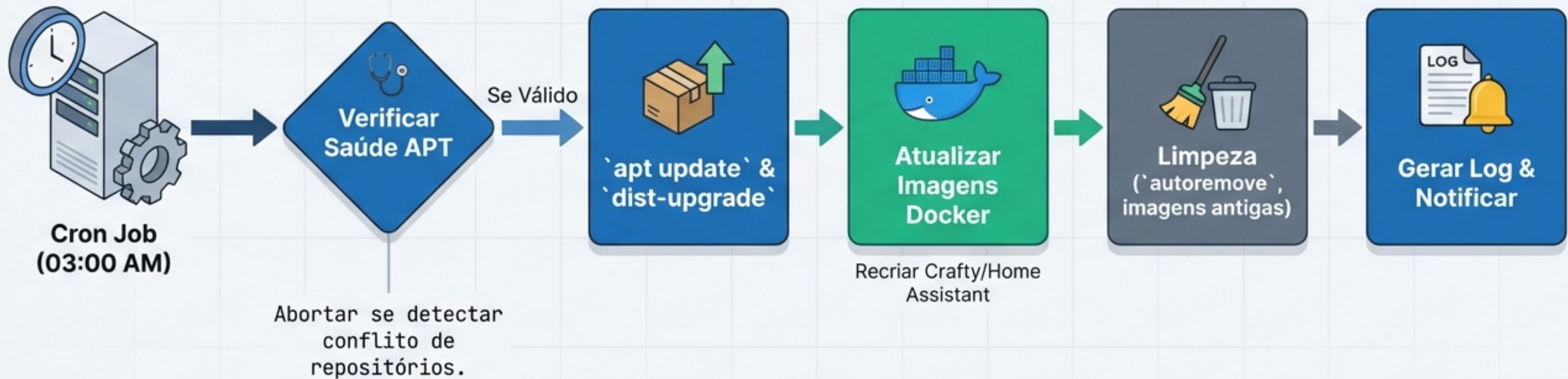


Node.js: Versão LTS gerenciada via NVM.



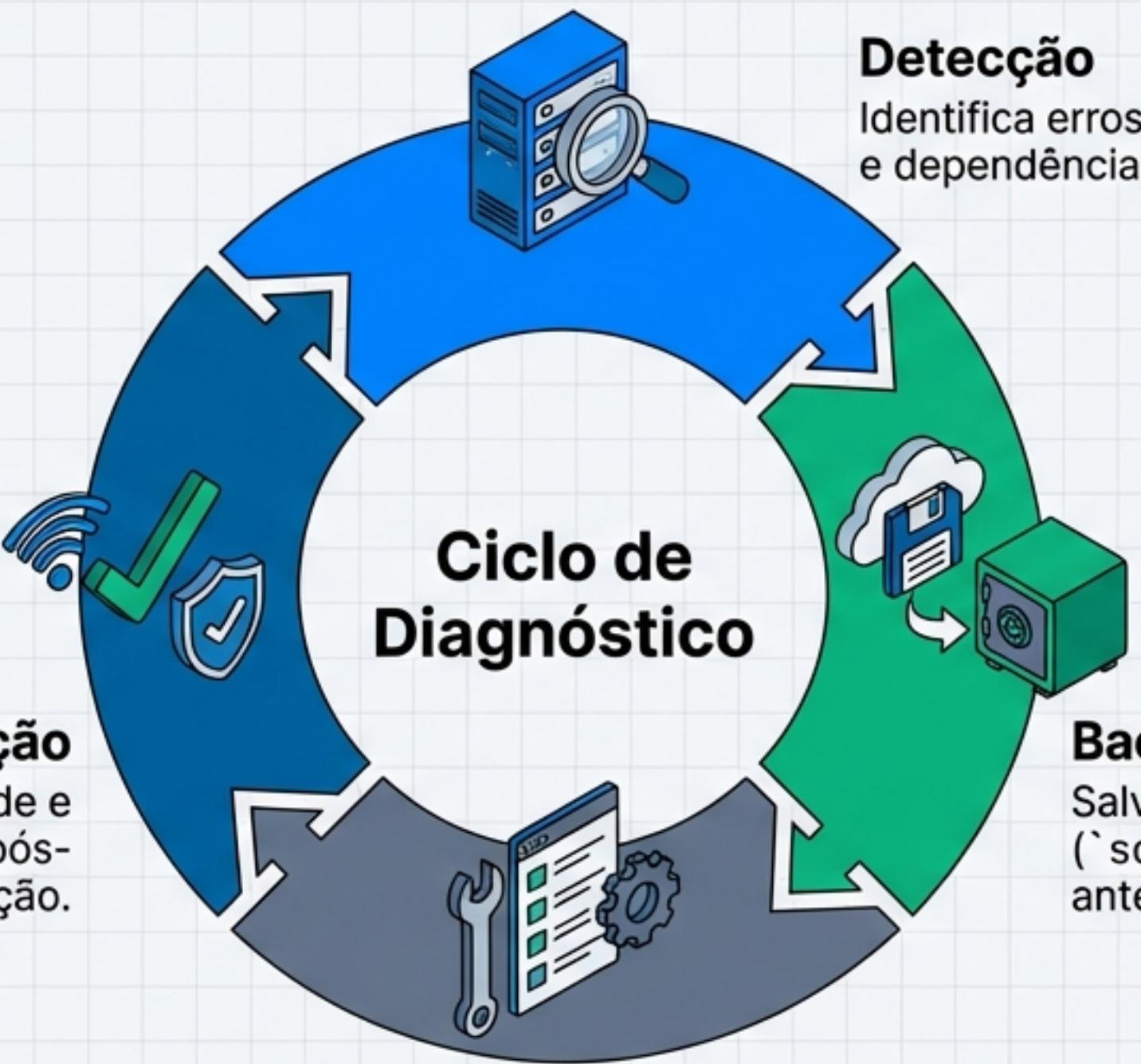
GitHub Tools: CLI (`gh`) e Copilot pré-configurados.

Fase 2: Atualização Contínua e Inteligente



Robustez e Auto-Cura (Self-Healing)

Ferramenta: `diagnostico_apt.sh`



Validação

Testa conectividade e integridade pós-correção.

Correção

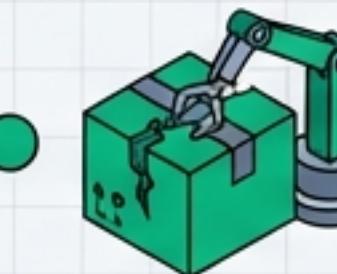
Remove duplicatas e corrige repositórios "Signed-By".

Detecção

Identifica erros de chaves GPG e dependências quebradas.



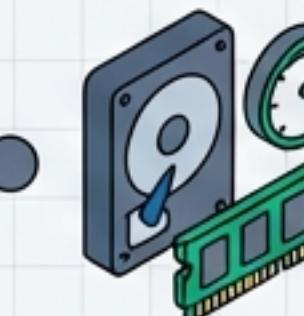
Detecção de Conflitos GPG



Correção automática de pacotes quebrados

Backup

Salva listas de fontes (`sources.list`) antes de alterar.



Verificação de espaço em disco e memória

Visibilidade Total: Logs e Relatórios

Log Bruto (Terminal)

```
[2024-05-15 03:05:01] Starting process...
[32mSUCCESS[0m: Base system updated.
[2024-05-15 03:05:45] 0;93mCONTAINER: crafty [0m
[2024-05-15 02:06:02] \033[1;31mERROR\033[0m: Port 8000
[2024-05-15 02:06:20] [S2wSUCCESS[0m: Container haos
[2024-05-15 03:06:45] Done. Exiting.
```

`converter_log_md.sh`



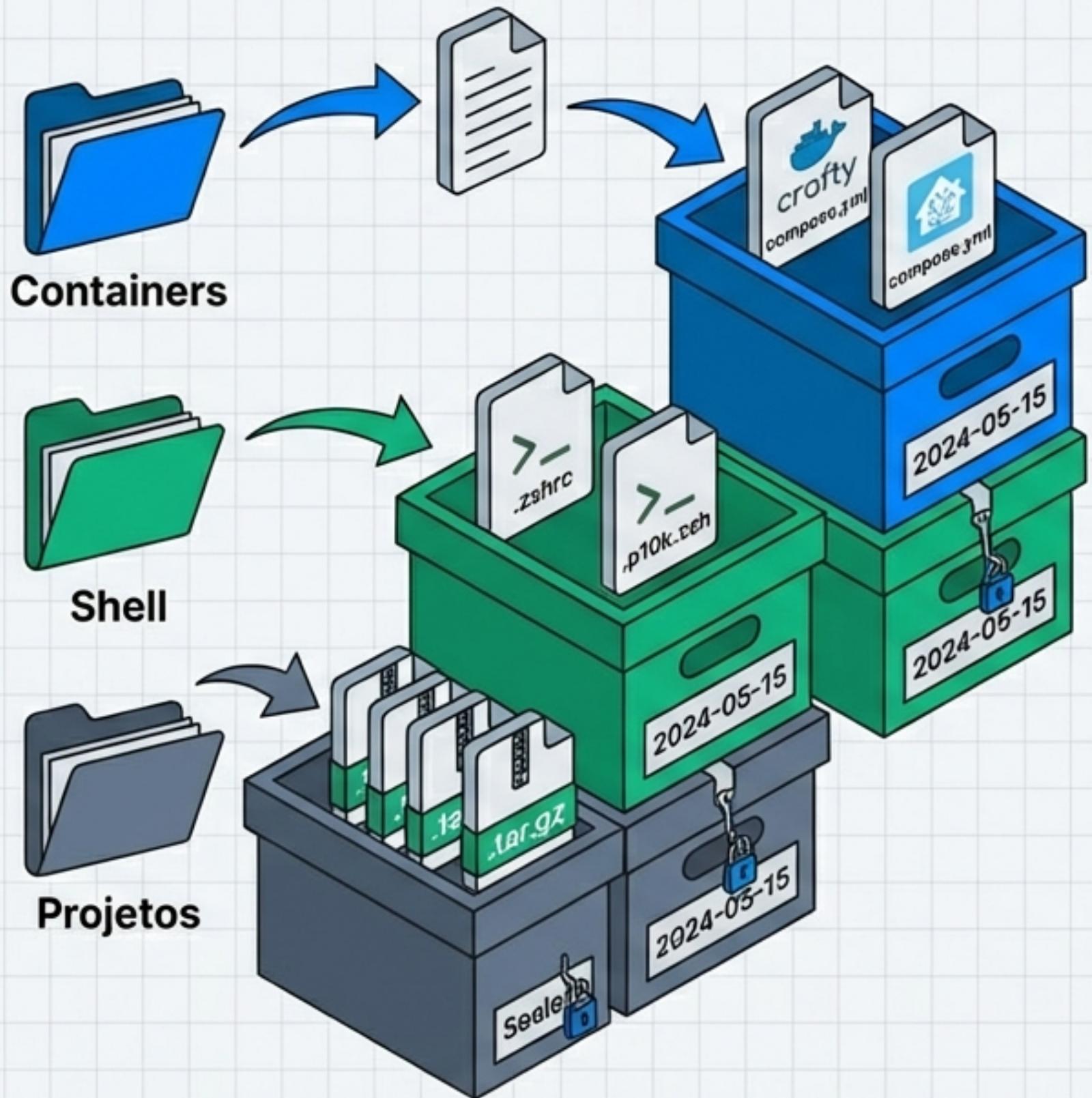
Relatório Markdown

Relatório de Atualização

SUCESSO: Sistema Base

Container ID	Status	Ports
crafty	Up 2 hours	8000
haos	Up 2 hours	8123

Estratégia de Backup Automatizada



Versionamento Inteligente

O sistema mantém um histórico de versões anteriores na pasta `backups/`, permitindo rollback rápido em caso de falha de configuração.

Executado automaticamente por `lib/backup_configs.sh` antes de cada atualização.

Garantia de Qualidade (QA)

107

Testes Automatizados (BATS)

	Instalação e Bootstrap		Sistema de Logging e Conversão Markdown
	Backup e Restauração		Detecção de duplicatas no APT

“Cada módulo é testado isoladamente para garantir que a integração do sistema permaneça estável.”

Personalização e Extensibilidade

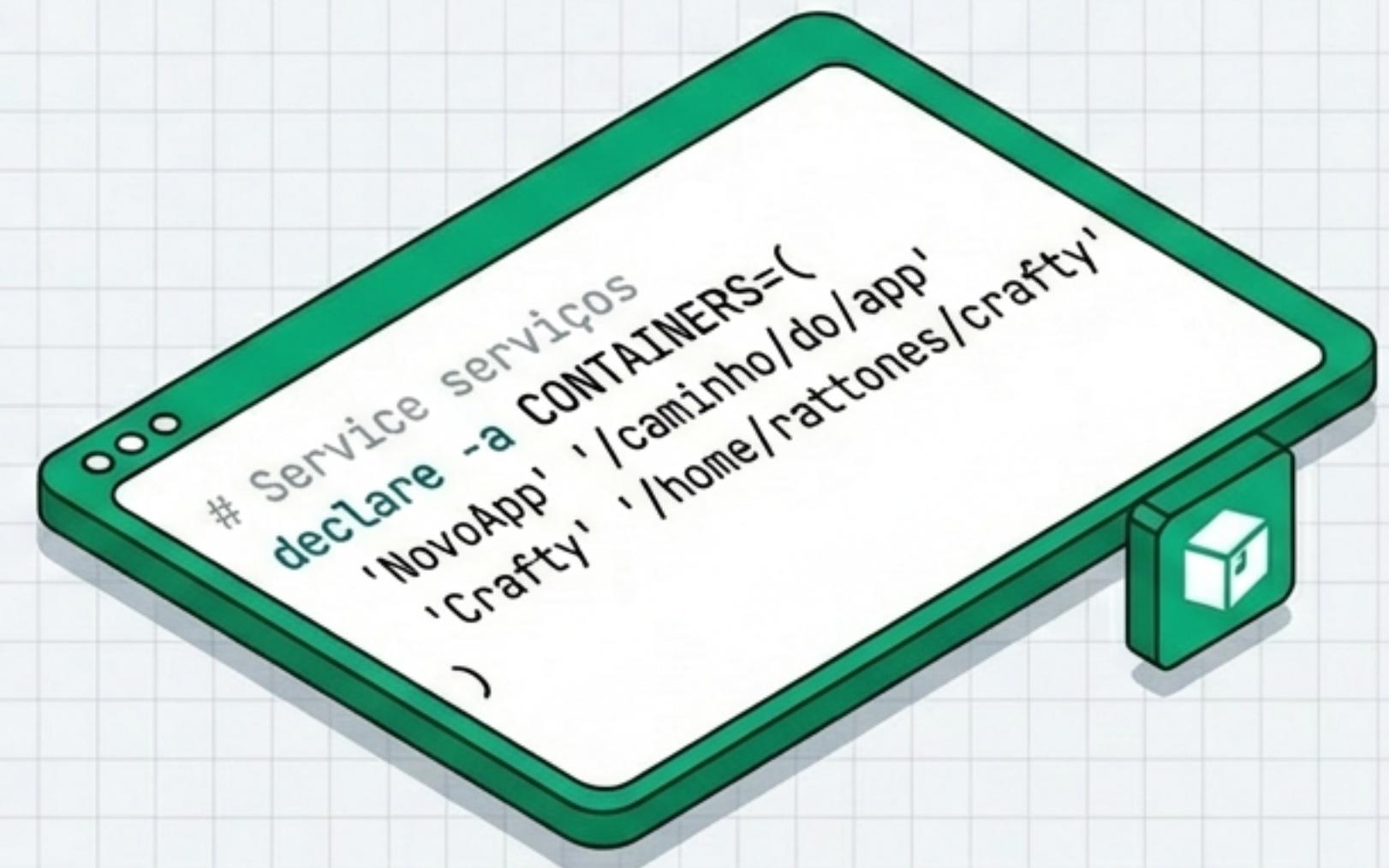
Bibliotecas Reutilizáveis

Importe funções do núcleo para seus próprios scripts:

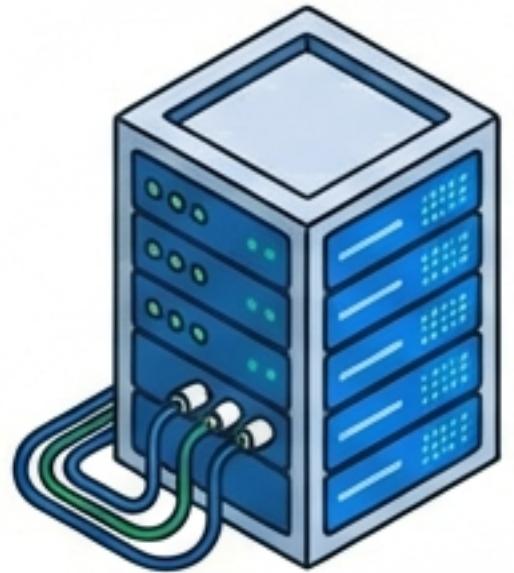


Configuração de Containers

Adicione novos serviços editando o array `CONTAINERS` no script principal:



Casos de Uso



SysAdmin

- Atualizações 'Set and forget'.
- Auditoria via relatórios Markdown.
- Gestão segura de chaves SSH.



Desenvolvedor

- Ambiente pronto em minutos.
- Git, Node, Docker pré-instalados.
- Ferramentas GitHub configuradas.



Home Lab

- Gerenciamento de Home Assistant.
- Servidores de Jogos (Crafty).
- Interface web (Cockpit) amigável.

Guia Rápido de Comandos

Instalação

```
./post-install.sh
```

Setup inicial completo.



Correção

```
sudo ./diagnostico_apt.sh
```

Detectar e corrigir erros do APT.



Manutenção

```
sudo ./atualizar_servidor.sh
```

Atualização segura de sistema e containers.



Utilidades

```
./lib/converter_log_md.sh
```

Gerar relatório legível de logs.



Comece Agora

github.com/ratttones/automacoes



Versão v1.1.0



Open Source



MIT License

“Automação inteligente para infraestrutura moderna.
Código limpo, testado e documentado.”

