

Plano Final Consolidado e Prompt para IA Desenvolvedora

Autor: Manus AI (Quality Assurance)

Data: 22 de Novembro de 2025

Versão: 4.0 (FINAL E DEFINITIVA)

PROMPT PARA A IA DESENVOLVEDORA

Plain Text

Você é uma IA desenvolvedora responsável pela reconstrução completa do Sistema VPS Admin.

DIRETRIZES FUNDAMENTAIS (NUNCA ESQUECER):

1. ****PDCA RIGOROSO:**** Nenhuma tarefa avança sem a anterior estar 100% completa e validada.
2. ****TUDO SEM INTERVENÇÃO MANUAL:**** Você deve fazer PR, commit, deploy, teste e tudo mais automaticamente.
3. ****NÃO COMPACTE, NÃO RESUMA, NÃO ECONOMIZE:**** Faça tudo completo. O importante é funcionar direito.
4. ****NÃO PARE. CONTINUE:**** Siga de onde parou e retorne onde falhou.
5. ****NÃO JULGUE O QUE É CRÍTICO:**** Tudo deve funcionar 100%. Não escolha partes críticas, faça TUDO.
6. ****SCRUM HIPERFRACIONADO:**** Siga os sprints na ordem, sem pular etapas.
7. ****BUSCAR NO GITHUB:**** Recupere a documentação de requisitos e especificações da versão anterior do GitHub.
8. ****RESTAURAR FUNCIONALIDADES:**** Vasculhe todas as versões até encontrar onde as funcionalidades listadas estavam 100% criadas. Recupere esses arquivos e coloque em produção.

ARQUIVOS DE REFERÊNCIA (SEMPRE CONSULTAR):

- `PLANO_FINAL_CONOLIDADO_E_PROMPT_IA.md` (este arquivo)
- `RELATORIO_COMPLETO_VPS_ADMIN_V2.md` (estado atual do sistema)
- `BACKLOG_COMPLETO_E_DEFINITIVO.md` (todas as funcionalidades)
- `PLANO_RECONSTRUCAO_FINAL_HIPERDETALHADO.md` (sprints detalhados)

CREDENCIAIS E ACESSOS:

Serviço Endereço Porta Usuário Senha
:--- :--- :--- :--- :---
Painel Admin `https://72.61.53.222/admin` 443 `admin@vps.local`

```
| `mcorpapp` |
| **SSH (Servidor)** | `72.61.53.222` | 22, 2222 | `root` | `Jm@D@KDPnw7Q` |
| **GitHub** | (a ser fornecido) | - | (a ser fornecido) | (a ser fornecido)
|
```

ORDEM DE EXECUÇÃO:

1. **Buscar no GitHub:** Clone o repositório e analise todas as versões (commits) para encontrar onde cada funcionalidade estava funcionando.
2. **Restaurar Funcionalidades:** Recupere os arquivos das versões funcionais e faça merge com a versão atual.
3. **Seguir os Sprints:** Execute os sprints na ordem definida no `PLANO_RECONSTRUCAO_FINAL_HIPERDETALHADO.md`.
4. **Validar Cada Etapa:** Após cada tarefa, execute o script de validação automatizado.
5. **Deploy Automático:** Faça deploy em produção após cada sprint validado.
6. **Nunca Parar:** Continue até que TODAS as funcionalidades estejam 100% funcionais.

COMO VALIDAR CADA ETAPA:

Após cada tarefa, execute um script de validação que:

1. Faz login no sistema
2. Navega para a funcionalidade desenvolvida
3. Executa as ações definidas nos critérios de aceitação
4. Verifica se o resultado esperado foi alcançado
5. Gera um relatório de sucesso ou falha

Se falhar, corrija e teste novamente. Não avance sem validação 100%.

EXEMPLO DE FLUXO DE TRABALHO:

```
```bash
1. Buscar versão funcional no GitHub
git log --all --grep="Email Domains" --oneline
git checkout <commit_hash> -- app/Http/Controllers/EmailDomainController.php

2. Fazer merge e resolver conflitos
git merge --no-commit

3. Testar localmente
python3 /opt/webserver/scripts/validate_sprint_2.py

4. Se passou, fazer commit
git add .
git commit -m "Sprint 2: Email Domains CRUD completo"

5. Deploy em produção
```

```

ssh root@72.61.53.222 -p 2222 "cd /opt/webserver/admin-panel && git pull &&
php artisan migrate && php artisan cache:clear"

6. Validar em produção
python3 /opt/webserver/scripts/validate_sprint_2_prod.py

7. Se passou, avançar para o próximo sprint

```

## REGRAS DE OURO:

- **✓ SEMPRE** consulte os arquivos de referência antes de cada sprint.
- **✓ SEMPRE** busque no GitHub antes de reescrever código.
- **✓ SEMPRE** valide antes de avançar.
- **✓ SEMPRE** faça deploy em produção após validação.
- **✗ NUNCA** pule etapas.
- **✗ NUNCA** assuma que algo funciona sem testar.
- **✗ NUNCA** pare antes de concluir 100%.

Agora, comece pelo Sprint 0 e siga até o final. Boa sorte!

### Plain Text

---

## 📦 BACKLOG COMPLETO E DEFINITIVO (CONSOLIDADO)

Este backlog consolida TODAS as funcionalidades identificadas nos relatórios anteriores + as funcionalidades da especificação de infraestrutura do servidor.

#### ### Épico 1: Autenticação e Segurança

ID	User Story	Prioridade	Status Atual
---	---	---	---
**1.1**	Como usuário, quero poder fazer login no sistema usando email e senha.	**Crítica**	✓ Funcional
**1.2**	Como usuário, quero poder fazer logout de forma segura.	**Crítica**	✓ Funcional
**1.3**	Como usuário, quero poder visualizar e atualizar as informações do meu perfil.	**Alta**	✓ Funcional
**1.4**	Como administrador, quero poder habilitar a autenticação de dois fatores (2FA).	**Média**	✗ Não existe

### ### Épico 2: Módulo de Email Domains (CRUD Completo)

ID	User Story	Prioridade	Status Atual
---	---	---	---
**2.1**	Como usuário, quero poder listar todos os domínios de email cadastrados.	**Crítica**    Funcional	
**2.2**	Como usuário, quero poder criar um novo domínio de email.	**Crítica**    Funcional	
**2.3**	Como usuário, quero poder editar um domínio de email existente.	Alta    Não existe	
**2.4**	Como usuário, quero poder excluir um domínio de email existente.	Alta    Funcional	

### ### Épico 3: Módulo de Email Accounts (CRUD Completo)

ID	User Story	Prioridade	Status Atual
---	---	---	---
**3.1**	Como usuário, quero poder listar todas as contas de email.	**Crítica**    Funcional	
**3.2**	Como usuário, quero poder criar uma nova conta de email.	**Crítica**    Funcional	
**3.3**	Como usuário, quero poder editar uma conta de email existente.	Alta    Não existe	
**3.4**	Como usuário, quero poder excluir uma conta de email existente.	Alta    Não existe	

### ### Épico 4: Módulo de Sites (CRUD Completo e Correção do Problema Histórico)

ID	User Story	Prioridade	Status Atual
---	---	---	---
**4.1**	Como usuário, quero poder listar todos os sites cadastrados.	**Crítica**    Funcional	
**4.2**	Como usuário, quero que a lista de sites seja atualizada IMEDIATAMENTE após a criação.	**Crítica**    **PROBLEMA HISTÓRICO (35 sprints)**	
**4.3**	Como usuário, quero poder criar um novo site.	**Crítica**    Não persiste	
**4.4**	Como usuário, quero poder editar as configurações de um site existente.	Alta    Não existe	
**4.5**	Como usuário, quero poder excluir um site existente.	Alta    Funcional	

### ### Épico 5: Módulo de Backups (Implementação Completa)

ID	User Story	Prioridade	Status Atual
---	---	---	---
**5.1**	Como usuário, quero poder listar todos os backups disponíveis.	Alta    Funcional	

**5.2**   Como usuário, quero poder criar um novo backup.   **Alta**   <span style="color: red;">●</span>
Erro 405
**5.3**   Como usuário, quero poder baixar um arquivo de backup.
**Média**   <span style="color: red;">●</span> Erro 405
**5.4**   Como usuário, quero poder restaurar o sistema a partir de um backup.   **Alta**   <span style="color: red;">●</span> Erro 405
**5.5**   Como administrador, quero poder agendar backups automáticos.
**Média**   <span style="color: red;">X</span> Não existe

#### ### Épico 6: Módulo de Logs (Implementação Completa)

ID	User Story	Prioridade	Status Atual
:---   :---   :---   :---			
**6.1**   Como usuário, quero poder listar os diferentes tipos de logs disponíveis.   **Média**   <span style="color: green;">✓</span> Funcional			
**6.2**   Como usuário, quero poder visualizar o conteúdo de um arquivo de log.   **Média**   <span style="color: red;">●</span> Erro 404			
**6.3**   Como usuário, quero poder limpar um arquivo de log.   **Baixa**   <span style="color: red;">●</span> Erro 404			

#### ### Épico 7: Módulo de Serviços (Implementação Completa)

ID	User Story	Prioridade	Status Atual
:---   :---   :---   :---			
**7.1**   Como usuário, quero poder listar os principais serviços e ver seu status.   **Alta**   <span style="color: green;">✓</span> Funcional			
**7.2**   Como usuário, quero poder reiniciar um serviço.   **Alta**   <span style="color: red;">●</span> Erro 404			
**7.3**   Como usuário, quero poder parar um serviço.   **Média**   <span style="color: red;">●</span> Erro 404			
**7.4**   Como usuário, quero poder iniciar um serviço.   **Média**   <span style="color: red;">●</span> Erro 404			

#### ### Épico 8: Dashboard e Monitoramento (NOVO)

ID	User Story	Prioridade	Status Atual
:---   :---   :---   :---			
**8.1**   Como usuário, quero ver um dashboard com o status geral do servidor (CPU, RAM, Disco).   **Alta**   <span style="color: red;">X</span> Não existe			
**8.2**   Como usuário, quero ver gráficos históricos do uso de recursos.   **Média**   <span style="color: red;">X</span> Não existe			
**8.3**   Como administrador, quero receber alertas por email se o uso de recursos ultrapassar 90%.   **Média**   <span style="color: red;">X</span> Não existe			

#### ### Épico 9: Gerenciamento de Servidor de Email (NOVO - da especificação)

ID	User Story	Prioridade	Status Atual
:---   :---   :---   :---			

**9.1**   Como administrador, quero poder configurar SPF, DKIM e DMARC para um domínio.	**Alta**   <span style="color:red">X</span> Não existe
**9.2**   Como administrador, quero poder visualizar a fila de emails (queue).	**Média**   <span style="color:red">X</span> Não existe
**9.3**   Como administrador, quero poder visualizar logs de spam e emails bloqueados.	**Média**   <span style="color:red">X</span> Não existe
**9.4**   Como administrador, quero poder gerenciar aliases de email.	**Média**   <span style="color:red">X</span> Não existe
**9.5**   Como usuário, quero poder acessar o webmail (Roundcube) diretamente do painel.	**Baixa**   <span style="color:red">X</span> Não existe

### ### Épico 10: Gerenciamento de Firewall (NOVO - da especificação)

ID	User Story	Prioridade	Status Atual
---	---	---	---
**10.1**   Como administrador, quero poder listar todas as regras de firewall ativas.	**Média**   <span style="color:red">X</span> Não existe		
**10.2**   Como administrador, quero poder adicionar uma nova regra de firewall.	**Média**   <span style="color:red">X</span> Não existe		
**10.3**   Como administrador, quero poder remover uma regra de firewall existente.	**Média**   <span style="color:red">X</span> Não existe		

### ### Épico 11: Gerenciamento de SSL/TLS (NOVO - da especificação)

ID	User Story	Prioridade	Status Atual
---	---	---	---
**11.1**   Como administrador, quero poder gerar certificados SSL Let's Encrypt para um domínio.	**Alta**   <span style="color:red">X</span> Não existe		
**11.2**   Como administrador, quero poder renovar certificados SSL automaticamente.	**Alta**   <span style="color:red">X</span> Não existe		
**11.3**   Como administrador, quero poder visualizar a data de expiração dos certificados.	**Média**   <span style="color:red">X</span> Não existe		

---

## ## SPRINTS HIPERFRACIONADOS (CONSOLIDADO)

### ### \*\*Sprint 0: Configuração do Ambiente e Recuperação do GitHub\*\*

**Objetivo:** Preparar o ambiente de desenvolvimento e recuperar funcionalidades da versão anterior do GitHub.

ID	Tarefa	Critérios de Aceitação
---	---	---
**0.1**   Clonar o repositório do GitHub.	O repositório está clonado localmente.	
**0.2**   Analisar o histórico de commits para identificar versões funcionais.	Lista de commits onde cada funcionalidade estava funcionando.	

**0.3**   Recuperar os arquivos das versões funcionais (Email Domains, Email Accounts).   Arquivos recuperados e salvos em `/tmp/recovered/`.	
**0.4**   Configurar o ambiente de desenvolvimento local.   `.`.env` configurado, dependências instaladas.	
**0.5**   Executar migrations e seeders.   Banco de dados criado e populado.	

### \*\*Sprint 1: Módulo de Autenticação e Perfil (Revisão)\*\*

(Já funcional, apenas validar)

### \*\*Sprint 2: Módulo de Email Domains (CRUD Completo)\*\*

**Objetivo:** Completar o CRUD de Email Domains, adicionando a funcionalidade de Edição.

ID   Tarefa   Critérios de Aceitação
:---   :---   :---
**2.1**   Implementar a rota `GET /admin/email/domains/{id}/edit`.   A rota existe e retorna HTTP 200.
**2.2**   Criar a view `email.domains.edit` com o formulário de edição.   A view é exibida com os dados do domínio.
**2.3**   Implementar o método `update()` no `EmailDomainController`.   O domínio é atualizado no banco.
**2.4**   Validar o CRUD completo (Create, Read, Update, Delete).   Todas as operações funcionam e são persistidas.

### \*\*Sprint 3: Módulo de Email Accounts (CRUD Completo)\*\*

**Objetivo:** Completar o CRUD de Email Accounts, adicionando as funcionalidades de Edição e Exclusão.

ID   Tarefa   Critérios de Aceitação
:---   :---   :---
**3.1**   Implementar a rota `GET /admin/email/accounts/{id}/edit`.   A rota existe e retorna HTTP 200.
**3.2**   Criar a view `email.accounts.edit` com o formulário de edição.   A view é exibida com os dados da conta.
**3.3**   Implementar o método `update()` no `EmailAccountController`.   A conta é atualizada no banco.
**3.4**   Implementar a rota `DELETE /admin/email/accounts/{id}`.   A conta é excluída do banco.
**3.5**   Validar o CRUD completo (Create, Read, Update, Delete).   Todas as operações funcionam e são persistidas.

### \*\*Sprint 4: Módulo de Sites (Correção do Problema Histórico de 35 Sprints)\*\*

**\*\*Objetivo:\*\*** Resolver definitivamente o problema de persistência de Sites.

ID	Tarefa	Critérios de Aceitação
---	---	---
**4.1**	**INVESTIGAÇÃO PROFUNDA:** Analisar o script shell `create_site.sh` linha por linha.   Identificar a causa da falha silenciosa.	
**4.2**	Implementar logging detalhado no script `create_site.sh`.   O script gera um arquivo de log com cada passo.	
**4.3**	Refatorar o método `store()` no `SiteController` para usar `ShellCommandService`.   O controller não executa mais o script diretamente.	
**4.4**	Garantir que a view `sites.index` não tenha cache ou filtro complexo.   A view apenas itera sobre os dados do controller.	
**4.5**	**TESTE DEFINITIVO:** Criar um novo site e verificar se aparece IMEDIATAMENTE na listagem.   O problema histórico está resolvido.	
**4.6**	Implementar as funcionalidades de Edição de sites.   O CRUD completo do módulo de Sites está funcional.	

### \*\*Sprint 5: Módulo de Backups (Implementação Completa)\*\*

**\*\*Objetivo:\*\*** Implementar todas as funcionalidades do módulo de Backups.

ID	Tarefa	Critérios de Aceitação
---	---	---
**5.1**	Implementar a rota `POST /admin/backups/create`.   A rota existe e retorna HTTP 200.	
**5.2**	Implementar o método `create()` no `BackupController`.   Um novo backup é criado e listado.	
**5.3**	Implementar a rota `GET /admin/backups/download/{id}`.   O arquivo de backup é baixado.	
**5.4**	Implementar a rota `POST /admin/backups/restore/{id}`.   O sistema é restaurado a partir do backup.	
**5.5**	Implementar agendamento de backups automáticos (cron).   Backups são criados automaticamente.	

### \*\*Sprint 6: Módulo de Logs (Implementação Completa)\*\*

**\*\*Objetivo:\*\*** Implementar todas as funcionalidades do módulo de Logs.

ID	Tarefa	Critérios de Aceitação
---	---	---
**6.1**	Implementar a rota `GET /admin/logs/view/{type}`.   O conteúdo do log é exibido.	
**6.2**	Implementar a rota `POST /admin/logs/clear/{type}`.   O arquivo de log é limpo.	

### \*\*Sprint 7: Módulo de Serviços (Implementação Completa)\*\*

**\*\*Objetivo:\*\*** Implementar todas as funcionalidades do módulo de Serviços.

ID	Tarefa	Critérios de Aceitação
7.1	Implementar a rota `POST /admin/services/restart/{service}`.	O serviço é reiniciado.
7.2	Implementar a rota `POST /admin/services/stop/{service}`.	O serviço é parado.
7.3	Implementar a rota `POST /admin/services/start/{service}`.	O serviço é iniciado.

### \*\*Sprint 8: Dashboard e Monitoramento (NOVO)\*\*

**\*\*Objetivo:\*\*** Implementar o dashboard com métricas em tempo real.

ID	Tarefa	Critérios de Aceitação
8.1	Implementar a coleta de métricas de CPU, RAM e Disco.	As métricas são coletadas via script shell.
8.2	Criar a view `dashboard.index` com cards de métricas.	dashboard exibe as métricas em tempo real.
8.3	Implementar gráficos históricos usando Chart.js.	Os gráficos são exibidos corretamente.
8.4	Implementar sistema de alertas por email.	Alertas são enviados quando o uso ultrapassa 90%.

### \*\*Sprint 9: Gerenciamento de Servidor de Email (NOVO)\*\*

**\*\*Objetivo:\*\*** Implementar funcionalidades de gerenciamento do servidor de email.

ID	Tarefa	Critérios de Aceitação
9.1	Implementar configuração de SPF, DKIM e DMARC.	Os registros DNS são gerados e exibidos.
9.2	Implementar visualização da fila de emails (queue).	A fila é exibida com detalhes.
9.3	Implementar visualização de logs de spam.	Os logs são exibidos e filtráveis.
9.4	Implementar gerenciamento de aliases de email.	Aliases podem ser criados, editados e excluídos.

### \*\*Sprint 10: Gerenciamento de Firewall (NOVO)\*\*

**\*\*Objetivo:\*\*** Implementar funcionalidades de gerenciamento do firewall.

ID	Tarefa	Critérios de Aceitação

:---   :---   :---
**10.1**   Implementar listagem de regras de firewall (UFW).   As regras são listadas corretamente.
**10.2**   Implementar adição de novas regras de firewall.   Novas regras são criadas e aplicadas.
**10.3**   Implementar remoção de regras de firewall.   Regras são removidas e aplicadas.

#### ### \*\*Sprint 11: Gerenciamento de SSL/TLS (NOVO)\*\*

\*\*Objetivo:\*\* Implementar funcionalidades de gerenciamento de certificados SSL.

ID   Tarefa   Critérios de Aceitação
:---   :---   :---
**11.1**   Implementar geração de certificados Let's Encrypt.   Certificados são gerados via Certbot.
**11.2**   Implementar renovação automática de certificados.   Certificados são renovados automaticamente.
**11.3**   Implementar visualização de data de expiração.   A data de expiração é exibida para cada domínio.

---

#### ## CRITÉRIOS DE ACEITAÇÃO GLOBAIS

Para que um sprint seja considerado concluído, TODOS os seguintes critérios devem ser atendidos:

1. \*\*Código Commitado:\*\* Todas as alterações foram commitadas no GitHub com mensagens claras.
2. \*\*Deploy em Produção:\*\* O código foi deployado no servidor de produção (`72.61.53.222`).
3. \*\*Testes Automatizados Passaram:\*\* O script de validação automatizado foi executado e retornou sucesso.
4. \*\*Validação Manual:\*\* Uma validação manual foi feita via browser para confirmar o funcionamento.
5. \*\*Documentação Atualizada:\*\* A documentação do projeto foi atualizada com as novas funcionalidades.

---

#### ## CONCLUSÃO

Este é o plano mais completo e detalhado possível para a reconstrução do Sistema VPS Admin. Ele consolida todo o conhecimento adquirido ao longo de 35 sprints de testes, incorpora todas as funcionalidades da especificação de infraestrutura do servidor e fornece um roteiro claro e acionável para a IA

desenvolvedora.

\*\*A IA desenvolvedora deve seguir este plano à risca, sem pular etapas, sem resumir e sem parar até que TODAS as funcionalidades estejam 100% funcionais.\*\*

\*\*Boa sorte e mãos à obra!\*\* 