

PROCEDIMENTO OPERACIONAL

MÉTODO DE DESENVOLVIMENTO DE SITES

SOP-010

Tratar Auditoria SEO, AEO e GEO

VERSÃO

v1.1 FINAL

DATA

11/09/2026

STATUS

FINAL — APROVADO

PROCEDIMENTO OPERACIONAL

SOP-010

Tratar Auditoria SEO, AEO e GEO

VERSÃO	v1.1 FINAL
DATA	11/09/2026
STATUS	FINAL — APROVADO
MANUAL PROPRIETÁRIO	Manual 6 — Testes, QA, Debugging, Segurança e SEO
MANUAIS RELACIONADOS	Manual 3 — Construção Visual; Manual 5 — Desenvolvimento com Codex, Lovable e GitHub; Manual 7 — Publicação, Entrega e Pós-Publicação; Manual 8 — Boas Práticas, Anti-Patterns, Práticas Depreciadas e Casos Reais
ESCOPO	AS-IS — interpretação, correção e validação de auditorias SEO/AEO/GEO no fluxo atual WSS
CAMADA NORMATIVA RELACIONADA	WSSPadraoTecnicoOperacionalSEOAEOGEOv0.2FINAL

NAVEGAÇÃO DO DOCUMENTO

Sumário

Estrutura do documento canônico v1.1 FINAL.

1. Finalidade	7
2. Gatilho — quando usar	7
3. Quando NÃO usar	7
4. Papéis	8
5. Pré-condições	9
6. Inputs	9
PARTE I — NÃO TRATAR A AUDITORIA COMO VERDADE AUTOMÁTICA	11
7. Princípio central	12
8. Passo 1 — Identificar origem e data do relatório	12
9. Passo 2 — Confirmar o estado atual	12
10. Passo 3 — Reproduzir cada achado relevante	13
PARTE II — CLASSIFICAR O ACHADO	14
11. Passo 4 — Classificar cada item	15
12. Registrar classificação antes de editar	15
13. Relatório baseado em build anterior	15
PARTE III — ENTENDER O QUE A FERRAMENTA ESTÁ ENXERGANDO	16
14. Passo 5 — Determinar a representação analisada	17
15. Caso real Daniel P	17
16. Passo 6 — Comparar UI × HTML inicial	17
17. Falso positivo × problema arquitetural	17
PARTE IV — SEO TÉCNICO AS-IS	19
18. Passo 7 — Revisar metadados quando o achado os envolve	20
19. Passo 8 — Revisar domínio/origem	20
20. Passo 9 — Revisar dados estruturados	20
21. Passo 10 — Revisar breadcrumbs e links internos	21
22. Passo 11 — Revisar sitemap	21
23. Passo 12 — Revisar robots.txt	21

24. Passo 13 — Revisar noindex/nofollow	21
PARTE V — SSR / HTML INICIAL	23
25. Passo 14 — Confirmar conteúdo público no HTML inicial quando aplicável	24
26. NORMA TÉCNICA VIGENTE	24
27. AS-IS × NORMA	25
PARTE VI — BLOG	26
28. Passo 15 — Revisar listagem do Blog	27
29. Passo 16 — Revisar artigo individual	27
30. Norma de Blog	27
PARTE VII — AEO / GEO NO AS-IS	28
31. AEO/GEO nas auditorias atuais	29
32. Não universalizar checklist Daniel P	29
PARTE VIII — SEOMATOR	31
33. Passo 17 — Ler o score como indicador, não como gate	32
34. Não criar threshold universal	32
35. Passo 18 — Decompor o relatório	32
PARTE IX — CONTEÚDO E VALIDAÇÃO HUMANA	33
36. Passo 19 — Identificar itens que dependem do cliente	34
37. Não declarar revisão humana inexistente	34
38. Conteúdo YMYL/sensível	34
PARTE X — DEFINIR O DELTA REAL	35
39. Passo 20 — Consolidar itens que precisam de alteração	36
40. Passo 21 — Ordenar correções por dependência	36
PARTE XI — IMPLEMENTAR	37
41. Passo 22 — Corrigir somente o delta	38
42. Preservar o que já está correto	38
43. Mudança localizada	38
PARTE XII — VALIDAR A CORREÇÃO	39
44. Passo 23 — Executar validação dirigida	40
45. Não transformar a lista em suíte universal	40
46. Passo 24 — Validar metadados	40

47. Passo 25 — Validar HTML inicial	40
48. Passo 26 — Validar sitemap/robots	41
49. Passo 27 — Validar links/breadcrumbs	41
PARTE XIII — REGRESSÃO	42
50. Passo 28 — Revalidar áreas relacionadas	43
51. Não quebrar o site para subir score	43
PARTE XIV — ACHADOS DE SEGURANÇA EM AUDITORIAS MISTAS	44
52. SEO/GEO pode trazer item de segurança	45
PARTE XV — STATUS DA AUDITORIA	46
53. Passo 29 — Consolidar resultado	47
54. Não usar “100%” sem critério sustentado	47
PARTE XVI — STOP CONDITION	48
55. Condição de parada	49
56. Checklist mínimo	49
57. O que NÃO é necessário para encerrar	49
PARTE XVII — CONTINGÊNCIAS	51
58. Auditoria diz que elemento está ausente, mas UI mostra	52
59. Relatório pede item já resolvido	52
60. Auditoria usa domínio preview	52
61. Score baixo por conteúdo não validado	52
62. Ferramenta sugere llms.txt	52
63. Crawler não vê conteúdo CSR	53
64. Correção SEO quebra componente/rota	53
PARTE XVIII — AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO	54
65. Não inventar	55
PARTE XIX — NORMA TÉCNICA VIGENTE	56
66. Arquitetura SSR-first / híbrida	57
67. CSR permitido para interação	57
68. Blog normativo	57
69. Áreas privadas	58
PARTE XX — ANTI-PATTERNS	59

70. Executar relatório antigo sem conferir main	60
71. Corrigir H1 que já existe sem entender o crawler	60
72. Transformar score em meta absoluta	60
73. Universalizar checklist de um projeto	60
74. Inventar revisão clínica	60
75. Adicionar schema sem correspondência real	60
76. Melhorar SEO sacrificando arquitetura/UX	60
PARTE XXI — OUTPUT	61
77. Saída esperada	62
PARTE XXII — PRÓXIMO PASSO	63
78. Encaminhamento	64
79. Referências	64

1. Finalidade

Tratar uma auditoria de SEO, AEO ou GEO como **fonte de evidência a ser verificada**, e não como ordem automática de alteração.

Fluxo principal:

```
relatório / auditoria
→ confirmar versão analisada
→ reproduzir no estado atual
→ classificar achado
→ separar limitação da ferramenta de delta real
→ identificar camada
→ corrigir somente o necessário
→ validar tecnicamente
→ registrar pendências humanas/normativas
→ encerrar
```

Este SOP documenta o **AS-IS de tratamento de auditorias**.

Ele não cria um novo padrão SEO/AEO/GEO.

2. Gatilho — quando usar

Usar quando houver:

- relatório de SEOmator;
- auditoria produzida por Lovable;
- auditoria produzida por Codex;
- checklist interno;
- relatório SEO/AEO/GEO;
- achado técnico de indexação;
- pedido de correção baseado em build/versão anterior;
- revisão técnica antes de publicação.

3. Quando NÃO usar

Não usar como substituto de:

- construção visual;
- QA geral → SOP-008;
- debugging geral → SOP-009;
- segurança;

- publicação → SOP-012;
- definição de estratégia editorial;
- criação de nova norma SEO/AEO/GEO.

Regra

Se a auditoria revelar bug técnico localizado:

usar SOP-009 para o diagnóstico detalhado.

4. Papéis

4.1 Desenvolvedor

Responsável por:

- interpretar o relatório;
- comparar com o estado atual;
- reproduzir achados;
- separar delta real de falso positivo/limitação;
- corrigir;
- validar;
- registrar dependências humanas.

4.2 Codex

Pode apoiar:

- inspeção do código;
- HTML SSR;
- build;
- links;
- schemas;
- metadados;
- sitemap;
- robots;
- validações dirigidas.

4.3 Lovable

Pode apoiar:

- análise;

- configuração;
- alterações integradas;
- backend/SSR quando aplicável.

4.4 SEOmator

É ferramenta externa de auditoria utilizada no AS-IS.

Seu resultado é:

input de investigação

e não autoridade automática.

4.5 Cliente / WSS

Pode ser necessário para validar:

- conteúdo factual;
- revisão clínica;
- autoria;
- fontes;
- dados institucionais;
- decisões editoriais.

5. Pré-condições

Antes de alterar:

1. relatório/auditoria disponível;
2. projeto/versão atual acessível;
3. branch/estado atual conhecidos;
4. domínio/origem relevantes conhecidos quando aplicável;
5. capacidade de reproduzir o achado;
6. norma técnica aplicável identificada quando necessário.

6. Inputs

Podem incluir:

- relatório SEOmator;
- plano Lovable/Codex;
- checklist AEO/GEO;

- robots.txt ;
 - sitemap;
 - HTML inicial;
 - metadados;
 - schemas;
 - rotas;
 - Blog;
 - Search Console quando aplicável;
 - domínio oficial;
 - norma técnica vigente;
 - versão/commit analisado.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE I — NÃO TRATAR A AUDITORIA COMO VERDADE AUTOMÁTICA

7. Princípio central

No AS-IS observado:

```
auditoria
→ estado atual
→ classificação
→ delta real
→ correção
→ validação
```

Não:

```
auditoria
→ executar tudo cegamente
```

8. Passo 1 — Identificar origem e data do relatório

Registrar:

- ferramenta;
- projeto;
- versão/build;
- momento aproximado;
- ambiente analisado.

Objetivo

Saber se o relatório ainda representa o projeto atual.

9. Passo 2 — Confirmar o estado atual

Abrir:

- código atual;
- `main` ou estado vigente;
- preview/ambiente correto.

Regra

Antes de corrigir:

comparar o pedido com a implementação atual.

10. Passo 3 — Reproduzir cada achado relevante

Para cada item:

- verificar se existe;
- localizar evidência;
- confirmar impacto.

Não confiar apenas no texto do relatório.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE II — CLASSIFICAR O ACHADO

11. Passo 4 — Classificar cada item

Usar as categorias AS-IS observadas:

A. Ainda presente

O problema existe no estado atual.

B. Parcialmente resolvido

Existe implementação, mas incompleta.

C. Já resolvido

O relatório descreve algo que já foi corrigido.

D. Não reproduzido

A ferramenta apontou algo que não é confirmável no estado atual.

12. Registrar classificação antes de editar

Essa classificação evita:

- retrabalho;
 - regressão;
 - edição baseada em histórico antigo.
-

13. Relatório baseado em build anterior

Caso conhecido:

- relatório pede correção;
- `main` já contém solução;
- alteração seria redundante.

Ação

Classificar como:

já resolvido

e não refazer.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE III — ENTENDER O QUE A FERRAMENTA ESTÁ ENXERGANDO

14. Passo 5 — Determinar a representação analisada

Perguntar:

- crawler lê HTML inicial?
- executa JavaScript?
- analisa DOM após hidratação?
- usa URL preview ou domínio oficial?
- acessa conteúdo autenticado?
- recebeu resposta SSR?

15. Caso real Daniel P

SEOmator relatou, entre outros:

- H1 ausente;
- imagens ausentes;
- schema ausente.

A UI mostrava elementos que o crawler não estava enxergando da mesma forma.

Lição

UI visível no navegador não prova que o conteúdo está presente no HTML entregue ao crawler.

16. Passo 6 — Comparar UI × HTML inicial

Quando houver contradição:

1. abrir página;
2. confirmar elemento visual;
3. inspecionar HTML inicial/SSR;
4. comparar o que o crawler recebe.

17. Falso positivo × problema arquitetural

Se a ferramenta não enxerga algo:

Pode ser

- limitação da ferramenta;
- configuração de crawl;
- ambiente incorreto;
- JavaScript client-side;
- problema real de renderização/indexabilidade.

Não decidir pela aparência

Investigar a camada real.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IV — SEO TÉCNICO AS-IS

18. Passo 7 — Revisar metadados quando o achado os envolve

Itens observados em auditorias reais:

- `title`;
 - meta description;
 - canonical;
 - `og:title`;
 - `og:description`;
 - `og:url`;
 - `og:image`;
 - Twitter Cards.
-

19. Passo 8 — Revisar domínio/origem

Confirmar:

- domínio oficial;
- preview;
- URLs absolutas;
- canonical;
- sitemap;
- OG.

Problema conhecido

Canonical editorial antigo pode apontar para preview.

Não presumir que a origem atual está correta sem verificar.

20. Passo 9 — Revisar dados estruturados

Auditorias reais verificaram:

- `Physician`;
- `WebSite`;
- `Blog`;
- `BlogPosting`;
- `BreadcrumbList`;
- outros schemas aplicáveis.

Regra

Schema precisa corresponder ao conteúdo real.

Não adicionar tipo apenas para elevar score.

21. Passo 10 — Revisar breadcrumbs e links internos

Quando aplicável:

- breadcrumbs;
- navegação contextual;
- links internos no Blog;
- links entre conteúdos relacionados.

22. Passo 11 — Revisar sitemap

Verificar:

- existência;
- URLs esperadas;
- domínio correto;
- ausência de rotas indevidas;
- artigos publicados quando a arquitetura prevê.

23. Passo 12 — Revisar `robots.txt`

Verificar:

- regras atuais;
- sitemap declarado quando aplicável;
- áreas privadas;
- bloqueios não intencionais.

24. Passo 13 — Revisar `noindex/nofollow`

Quando houver rotas que não devem indexar:

- login;

- Admin;
- dashboards;
- rotas conceituais internas;
- outras áreas privadas.

Importante

A aplicabilidade depende da rota/projeto.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE V — SSR / HTML INICIAL

25. Passo 14 — Confirmar conteúdo público no HTML inicial quando aplicável

Para páginas públicas indexáveis, verificar se o conteúdo principal aparece no HTML entregue inicialmente.

Itens importantes:

- H1;
- conteúdo principal;
- links;
- metadados;
- schema quando aplicável.

26. NORMA TÉCNICA VIGENTE

A autoridade canônica atual para SEO, AEO e GEO é:

- WSS_Padrao_Tecnico_Operacional_SEO_AEO_GEO_v0.2_FINAL .

O antigo WSS_Padrao_Tecnico_Arquitetura_SEO_v0.1 permanece apenas como baseline histórico/proveniência, não como autoridade autônoma vigente.

O padrão vigente mantém abordagem:

SSR-first / híbrida

para páginas públicas indexáveis.

Classificações normativas

Os requisitos do padrão podem ser classificados como:

- Obrigatória;
- Recomendada;
- Condicional;
- Experimental;
- Não adotada;
- Sem decisão.

A classificação deve ser preservada; recomendação não deve ser silenciosamente convertida em obrigação.

Regra normativa

Páginas públicas devem preservar, conforme a classificação aplicável:

- HTML completo por URL;
- conteúdo principal;

- headings;
- links;
- metadados;
- dados estruturados aplicáveis;
- status HTTP correto;

sem depender do cliente para materializar o conteúdo principal.

Também devem ser considerados, no recorte aplicável:

- staging/preview não indexável e fora do sitemap oficial;
- origem oficial de produção para canonical e demais URLs absolutas;
- Google Search Console configurado na publicação;
- Bing Webmaster Tools como recomendado;
- GA4 como recomendado, não obrigatório universal;
- `llms.txt` como não obrigatório;
- política padrão de crawlers de treinamento ainda sem decisão final;
- ferramentas pagas sem escolha definitiva.

27. AS-IS × NORMA

Não afirmar que todo projeto histórico já cumpre essa arquitetura.

Regra

```
AS-IS
= o que o projeto realmente faz

NORMA
= o que a arquitetura vigente exige
```

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VI — BLOG

28. Passo 15 — Revisar listagem do Blog

Quando o Blog entra na auditoria:

- conteúdo publicado;
 - cards/títulos;
 - links;
 - metadados;
 - canonical;
 - HTML inicial;
 - domínio.
-

29. Passo 16 — Revisar artigo individual

Verificar:

- slug;
 - H1;
 - corpo;
 - metadados;
 - canonical;
 - Open Graph;
 - schema;
 - breadcrumbs;
 - links internos;
 - comportamento para slug inexistente.
-

30. Norma de Blog

A norma técnica vigente define Blog SSR para novos projetos públicos.

Isso não deve ser apresentado como se fosse prática histórica universal de todo legado.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VII — AEO / GEO NO AS-IS

31. AEO/GEO nas auditorias atuais

Podem envolver:

Visibilidade técnica

- HTML acessível;
- headings;
- structured data;
- robots;
- sitemap;
- crawlers;
- canonical.

Estrutura de conteúdo

- FAQ;
- resumo;
- TL;DR;
- headings interrogativos;
- entidades;
- variações semânticas;
- links internos.

Autoridade

- autoria/revisão;
- E-E-A-T;
- fontes;
- entidade/schema;
- sameAs .

32. Não universalizar checklist Daniel P

O projeto Daniel P possui checklist interno real com grupos:

- AEO;
- GEO;
- SEO;
- E-E-A-T.

Exemplos observados:

AEO

- FAQ;
- "Em resumo";
- TL;DR;
- headings em forma de pergunta.

GEO

- entidades relacionadas;
- variações semânticas;
- "Quem deve ler/agir";
- links internos.

SEO

- tamanho de título;
- meta description;
- palavra-chave;
- estrutura;
- contagem.

E-E-A-T

- revisor médico;
- última revisão.

Regra

Isso prova uma implementação real.

Não prova um checklist universal WSS.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VIII — SEOmator

33. Passo 17 — Ler o score como indicador, não como gate

Caso real Daniel P:

- auditoria GEO SEOmator;
- score 82/100.

Dimensões incluíam:

- AI Citability;
- Brand Authority;
- Platform Readiness.

34. Não criar threshold universal

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Não existe score mínimo SEOmator universal aprovado para todo projeto WSS no AS-IS.

Heurística pessoal declarada pelo Dev — não normativa

O Dev busca, como referência pessoal, chegar à faixa verde; considera 90 um resultado bom e 95+ uma meta desejada quando viável. Isso não é gate WSS, não é requisito de publicação/entrega e não autoriza corrigir cegamente o relatório para elevar score.

35. Passo 18 — Decompor o relatório

Separar:

1. achado;
2. interpretação;
3. recomendação;
4. implementação comprovada.

Regra

Plano sugerido pela ferramenta não equivale a implementação.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IX — CONTEÚDO E VALIDAÇÃO HUMANA

36. Passo 19 — Identificar itens que dependem do cliente

Exemplos:

- revisão clínica;
- título/formação;
- autoria;
- fatos;
- fontes;
- data de revisão;
- informação institucional.

37. Não declarar revisão humana inexistente

Caso real:

- revisão clínica dependia do cliente;
- auditoria não declarou falsamente como concluída.

Regra

Automação técnica não substitui validação factual.

38. Conteúdo YMYL/sensível

Quando o conteúdo é médico ou factual sensível:

- não inventar;
- não completar lacunas;
- preservar decisão humana.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE X — DEFINIR O DELTA REAL

39. Passo 20 — Consolidar itens que precisam de alteração

Depois da triagem:

Corrigir

- ainda presente;
- parcial.

Não tocar

- já resolvido;
- não reproduzido sem evidência.

Escalar

- conteúdo/decisão humana;
 - mudança normativa/estratégica;
 - escopo novo.
-

40. Passo 21 — Ordenar correções por dependência

Quando uma correção depende de outra:

- tratar suporte primeiro.

Exemplo:

- domínio/origem;
 - depois canonical/OG absoluto.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XI — IMPLEMENTAR

41. Passo 22 — Corrigir somente o delta

Aplicar mudanças dirigidas.

Evitar:

- reescrever SEO inteiro;
 - substituir implementação saudável;
 - adicionar requisito só porque apareceu em relatório externo.
-

42. Preservar o que já está correto

Caso real Guilherme:

- canonical já resolvido → preservado;
 - sitemap não tinha problema antigo → preservado.
-

43. Mudança localizada

Quando problema é:

- robots sem sitemap;
- OG parcial;
- schema parcial;

corrigir o item correspondente.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XII — VALIDAR A CORREÇÃO

44. Passo 23 — Executar validação dirigida

Auditorias reais utilizaram, conforme caso:

- ESLint direcionado;
- `git diff --check`;
- build;
- inspeção de HTML SSR;
- origem oficial simulada;
- navegação;
- teste de links.

45. Não transformar a lista em suíte universal

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Essas validações são exemplos reais.

Não existe uma combinação única obrigatória para toda auditoria.

46. Passo 24 — Validar metadados

Conferir, conforme alteração:

- title;
- description;
- canonical;
- OG;
- Twitter.

47. Passo 25 — Validar HTML inicial

Quando a correção envolve renderização/indexabilidade:

- inspecionar resposta inicial;
- confirmar conteúdo/metadados/schema.

48. Passo 26 — Validar sitemap/robots

Quando alterados:

- abrir;
 - conferir conteúdo;
 - conferir domínio;
 - conferir referência entre arquivos.
-

49. Passo 27 — Validar links/breadcrumbs

Quando alterados:

- navegar;
 - clicar;
 - confirmar destino.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIII — REGRESSÃO

50. Passo 28 — Revalidar áreas relacionadas

Exemplos:

Metadados compartilhados

Revisar mais de uma rota.

Helper de origem

Revisar canonical/OG/sitemap.

Blog

Revisar listagem + artigo.

Robots/noindex

Revisar áreas públicas e privadas relacionadas.

51. Não quebrar o site para subir score

Uma correção SEO/AEO/GEO não pode:

- prejudicar navegação;
 - quebrar SSR;
 - quebrar Blog;
 - expor área privada;
 - alterar conteúdo factual indevidamente.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIV — ACHADOS DE SEGURANÇA EM AUDITORIAS MISTAS

52. SEO/GEO pode trazer item de segurança

Exemplos:

- HTTPS;
- security headers;
- exposição;
- crawler access.

Limite

Este SOP não executa auditoria completa de segurança.

Se houver achado relevante:

- encaminhar à validação de segurança proporcional;
- não misturar correção superficial com autorização/PII.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XV — STATUS DA AUDITORIA

53. Passo 29 — Consolidar resultado

Para cada item, registrar uma situação:

- corrigido;
 - preservado porque já correto;
 - não reproduzido;
 - pendente de cliente;
 - pendente de decisão;
 - fora do escopo.
-

54. Não usar “100%” sem critério sustentado

Não declarar auditoria perfeita apenas porque:

- ferramentas não mostram mais alertas;
- score subiu.

O estado precisa refletir o que foi realmente validado.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVI — STOP CONDITION

55. Condição de parada

O SOP está concluído quando:

os achados relevantes da auditoria foram confrontados com o estado atual, classificados, os deltas reais foram corrigidos, as correções foram tecnicamente revalidadas no nível proporcional e os itens dependentes de cliente, norma ou decisão humana estão explicitamente registrados.

56. Checklist mínimo

Antes de encerrar:

- ☐ origem/data/versão da auditoria identificadas;
- ☐ estado atual conferido;
- ☐ achados relevantes reproduzidos;
- ☐ cada item classificado;
- ☐ UI × HTML inicial comparados quando necessário;
- ☐ domínio/origem conferidos quando aplicável;
- ☐ metadados conferidos quando aplicável;
- ☐ schemas conferidos quando aplicável;
- ☐ sitemap/robots conferidos quando aplicável;
- ☐ Blog conferido quando aplicável;
- ☐ delta real implementado;
- ☐ validação técnica executada;
- ☐ regressão dirigida executada;
- ☐ dependências humanas registradas;
- ☐ norma técnica mantida em camada separada.

57. O que NÃO é necessário para encerrar

Não exigir universalmente:

- score SEOMator específico;
- mesma checklist AEO/GEO em todo projeto;
- `llms.txt`;
- mesma quantidade de FAQ;
- mesma contagem de palavras;
- mesmo conjunto de schemas;

- auditoria externa paga;
 - implementação de toda recomendação da ferramenta.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVII — CONTINGÊNCIAS

58. Auditoria diz que elemento está ausente, mas UI mostra

1. confirmar UI;
 2. conferir HTML inicial;
 3. determinar modo de coleta;
 4. classificar limitação × problema real;
 5. corrigir somente se houver delta.
-

59. Relatório pede item já resolvido

Classificar:

já resolvido

e preservar.

60. Auditoria usa domínio preview

Reavaliar resultados dependentes de:

- canonical;
 - OG;
 - sitemap;
 - origem.
-

61. Score baixo por conteúdo não validado

Não inventar conteúdo.

Registrar dependência humana.

62. Ferramenta sugere `llms.txt`

Não tornar obrigatório apenas porque apareceu na auditoria.

Avaliar à luz da norma vigente e decisão WSS.

63. Crawler não vê conteúdo CSR

Investigar se:

- é conteúdo principal/indexável;
 - deveria estar no HTML inicial conforme norma;
 - é interação legítima CSR.
-

64. Correção SEO quebra componente/rota

Tratar como regressão.

Usar SOP-009.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVIII — AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO

65. Não inventar

O AS-IS não possui:

- score mínimo SEOmator universal;
 - ferramenta externa única obrigatória;
 - checklist AEO/GEO único para todos os projetos;
 - `llms.txt` universal obrigatório;
 - número universal de FAQs;
 - contagem universal de palavras;
 - conjunto fixo de schemas para todo site;
 - pipeline única de validação;
 - obrigação universal de usar auditoria paga;
 - regra de implementar toda recomendação de crawler.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIX — NORMA TÉCNICA VIGENTE

66. Arquitetura SSR-first / híbrida

O documento normativo vigente é:

- `WSS_Padrao_Tecnico_Operacional_SEO_AEO_GEO_v0.2_FINAL`.

Ele estabelece, para páginas públicas indexáveis e conforme a classificação aplicável:

- SSR como padrão;
- HTML completo e específico por URL;
- conteúdo principal no HTML inicial;
- metadados;
- canonical;
- dados estruturados aplicáveis;
- links rastreáveis;
- status HTTP adequado.

Para verificação de conformidade, consultar sem duplicar:

- `WSS_SEO_AEO_GEO_Matriz_Requisitos_Tecnicos_Parte2_v0.3`;
 - `WSS_SEO_AEO_GEO_Matriz_Schema_por_Tipo_Pagina_v0.3`;
 - `WSS_SEO_AEO_GEO_Checklist_PrePublicacao_v0.3`.
-

67. CSR permitido para interação

A norma admite CSR em:

- busca;
- filtros;
- formulários;
- Chat;
- Chatbot;
- CTA;
- áreas privadas.

Não deve substituir o conteúdo principal público.

68. Blog normativo

A norma vigente define:

- listagem SSR;
- artigo SSR;

- metadados próprios;
 - schema;
 - breadcrumbs;
 - 404 real;
 - publicação via Admin sem depender de rebuild para cada artigo.
-

69. Áreas privadas

Normativamente:

- noindex;
- fora do sitemap;
- autenticação adequada;
- autorização não baseada apenas em ocultação visual.

Decisões preservadas

Este SOP não decide:

- política padrão para crawlers de treinamento;
 - seleção definitiva de ferramenta paga;
 - obrigatoriedade universal futura do GA4;
 - fronteira entre entrega padrão e serviço recorrente.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XX — ANTI-PATTERNS

70. Executar relatório antigo sem conferir `main`

Pode reintroduzir problema.

71. Corrigir H1 que já existe sem entender o crawler

Pode atacar o sintoma errado.

72. Transformar score em meta absoluta

Ferramenta não substitui julgamento técnico.

73. Universalizar checklist de um projeto

Daniel P não define sozinho o padrão WSS.

74. Inventar revisão clínica

Revisão humana precisa existir de fato.

75. Adicionar schema sem correspondência real

Structured data não deve inventar entidade/conteúdo.

76. Melhorar SEO sacrificando arquitetura/UX

Correção não pode quebrar o produto.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XXI — OUTPUT

77. Saída esperada

Ao final:

Triagem

- auditoria comparada ao estado atual;
- itens classificados.

Correção

- somente deltas reais alterados;
- itens corretos preservados.

Validação

- metadados/HTML/schema/sitemap/robots/links revalidados conforme aplicabilidade;
- regressão proporcional realizada.

Governança

- pendências humanas explícitas;
 - recomendações não obrigatórias separadas;
 - norma técnica identificada em camada própria.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XXII — PRÓXIMO PASSO

78. Encaminhamento

Depois do SOP-010:

- bug/regressão → **SOP-009**;
 - feedback do cliente → **SOP-011**;
 - publicação final → **SOP-012**.
-

79. Referências

Manual proprietário

- WSS_Manual_06_Testes_QA_Debugging_Seguranca_SE0_v1.1_FINAL.md

Manuais relacionados

- WSS_Manual_03_Construcao_Visual_v1.1_FINAL.md ;
- WSS_Manual_05_Development_Codex_Lovable_GitHub_v1.1_FINAL.md ;
- WSS_Manual_07_Publicacao_Entrega_Pos_Publicacao_v1.1_FINAL.md ;
- WSS_Manual_08_Boas_Praticas_AntiPatterns_Depreciadas_Casos_Reais_v1.1_FINAL.md .

Baseline

- WSS_Arqueologia_Tecnica_ASIS_v0.3.md ;
- WSS_Matriz_Cobertura_ASIS_P0_P1_P2_v0.3_FINAL.md ;
- WSS_Handoff_Consolidacao_Manuais_SOPs_Skills_v0.2.md .

Evidência técnica principal

- WSS_VAL_003_Auditorias_SE0_AE0_GEO_Seguranca_v0.1.md .

Evidência de troubleshooting

- WSS_VAL_004_Troubleshooting_Estudos_de_Caso_v0.1.md .

Norma técnica relacionada

- WSS_Padrao_Tecnico_Operacional_SE0_AE0_GEO_v0.2_FINAL ;
 - WSS_Padrao_Tecnico_Arquitetura_SE0_v0.1 — baseline histórico/proveniência;
 - WSS_SE0_AE0_GEO_Matriz_Requisitos_Tecnicos_Parte2_v0.3 ;
 - WSS_SE0_AE0_GEO_Matriz_Schema_por_Tipo_Pagina_v0.3 ;
 - WSS_SE0_AE0_GEO_Matriz_Crawlers_Diretivas_v0.2 ;
 - WSS_SE0_AE0_GEO_Checklist_PrePublicacao_v0.3 .
-

Status: FINAL — APROVADO