

PROCEDIMENTO OPERACIONAL

MÉTODO DE DESENVOLVIMENTO DE SITES

SOP-008

Executar QA Visual, Funcional e Responsivo

VERSÃO

v1.1 FINAL

DATA

11/09/2026

STATUS

FINAL — APROVADO

PROCEDIMENTO OPERACIONAL

SOP-008

Executar QA Visual, Funcional e Responsivo

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 4 — Projeto-Base, Admin e Funcionalidades; Manual 7 — Publicação, Entrega e Pós-Publicação; Manual 8 — Boas Práticas, Anti-Patterns, Práticas Depreciadas e Casos Reais
ESCOPO	AS-IS — validação visual, responsiva e funcional antes da revisão final/publicação

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
7. Ferramentas e ambientes	10
PARTE I — PREPARAR A RODADA	11
8. Passo 1 — Confirmar versão e ambiente	12
9. Passo 2 — Definir escopo do QA	12
10. Passo 3 — Revisar Sitemap e módulos aplicáveis	12
PARTE II — QA VISUAL	14
11. Passo 4 — Percorrer página por página	15
12. Passo 5 — Observar fidelidade visual	15
13. Passo 6 — Verificar tipografia	15
14. Passo 7 — Verificar espaçamentos e alinhamentos	16
15. Passo 8 — Verificar imagens e assets	16
16. Passo 9 — Verificar efeitos/animações	16
PARTE III — QA RESPONSIVO	18
17. Três cenários principais do AS-IS	19
18. Passo 10 — Validar desktop	19
19. Passo 11 — Validar notebook	19
20. Passo 12 — Validar mobile físico	20
21. Passo 13 — Procurar scroll horizontal	20
22. AUSÊNCIA ATUAL DE PADRONIZAÇÃO — viewports intermediários	20
PARTE IV — NAVEGAÇÃO E ROTAS	22
23. Passo 14 — Testar navegação principal	23

24. Passo 15 — Testar CTAs	23
25. Passo 16 — Testar rotas públicas principais	23
26. 404	24
PARTE V — QA DE CTAS, FORMULÁRIOS E LEADS	25
27. Passo 17 — Identificar fluxos de captação	26
28. Passo 18 — Executar formulário ponta a ponta	26
29. Passo 19 — Conferir Lead	26
30. Passo 20 — Testar erros evidentes do formulário	27
PARTE VI — QA DE CHAT / CTA / WHATSAPP	28
31. Passo 21 — Testar Chat	29
32. Passo 22 — Testar WhatsApp	29
PARTE VII — QA DE CHATBOT	30
33. Quando Chatbot se aplica	31
34. Passo 23 — Revalidar fluxo básico do Chatbot	31
35. Teste dirigido adicional	31
PARTE VIII — QA DO ADMIN	32
36. Passo 24 — Abrir Admin	33
37. Passo 25 — Percorrer módulo por módulo	33
38. Passo 26 — Executar função essencial de cada módulo	33
PARTE IX — QA DE PERFIS E ACESSO	35
39. Passo 27 — Testar perfil Equipe/cliente	36
40. Passo 28 — Testar perfil Webstar	36
41. Limite	36
PARTE X — DADOS DE TESTE	37
42. Passo 29 — Usar dados claramente fictícios	38
43. Não limpar cedo demais	38
PARTE XI — ERROS, EMPTY STATES E CASOS NÃO COBERTOS	39
44. Empty states	40
45. Estados de erro	40
PARTE XII — USO DO CODEX NO QA	41
46. Passo 30 — Solicitar rodada ampla ao Codex quando útil	42

47. Não delegar decisão final	42
48. Build e typecheck	42
PARTE XIII — REGISTRAR PROBLEMAS	43
49. Passo 31 — Registrar cada falha reproduzível	44
50. Classificar por camada	44
PARTE XIV — CORREÇÃO E REVALIDAÇÃO	45
51. Passo 32 — Corrigir problemas localizados	46
52. Passo 33 — Encaminhar problema não trivial	46
53. Passo 34 — Revalidar exatamente o fluxo que falhou	46
PARTE XV — REGRESSÃO	47
54. Passo 35 — Revalidar áreas impactadas	48
55. Regressão visual	48
56. Regressão funcional	48
57. Regressão de perfil	48
PARTE XVI — REVISÃO HUMANA ADICIONAL	49
58. Revisão adicional em paralelo	50
PARTE XVII — PROCESSO SITES WSS / MARCO 8	51
59. Relação com o Marco 8	52
60. Processo ≠ SOP	52
PARTE XVIII — STOP CONDITION	53
61. Condição de parada	54
62. Checklist mínimo	54
63. O que NÃO é necessário para encerrar	54
PARTE XIX — CONTINGÊNCIAS	56
64. Visual correto em desktop e quebrado em mobile	57
65. CTA termina em 404	57
66. Formulário mostra sucesso mas Lead não aparece	57
67. Chat abre mas conversa não chega ao Admin	57
68. Admin abre para Webstar, mas não para Equipe	57
69. Efeito quebra somente em mobile	58
70. Codex aponta problema que o Dev não reproduz	58

PARTE XX — AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO	59
71. Não inventar	60
PARTE XXI — NORMA TÉCNICA VIGENTE	61
72. Normas complementam o QA	62
PARTE XXII — ANTI-PATTERNS	63
73. Validar só a Home	64
74. Validar só desktop	64
75. Validar formulário apenas visualmente	64
76. Validar Chat apenas pelo widget	64
77. Confiar em primeira saída do Codex	64
78. Corrigir sem revalidar	64
79. Testar só área corrigida quando componente é compartilhado	64
PARTE XXIII — OUTPUT	65
80. Saída esperada	66
PARTE XXIV — PRÓXIMO PASSO	67
81. Encaminhamento	68
82. Referências	68

1. Finalidade

Verificar se o site e sua camada operacional estão visualmente corretos, navegáveis e funcionalmente utilizáveis antes de avançar para revisão final, homologação e publicação.

Fluxo principal:

```
páginas suficientemente desenvolvidas
→ atualizar/abrir ambiente
→ teste visual
→ teste responsivo
→ teste funcional do site público
→ teste de formulários/Leads
→ teste de Chat/Chatbot quando aplicável
→ teste do Admin
→ teste de perfis quando aplicável
→ registrar problemas
→ corrigir/revalidar
→ executar regressão dirigida
→ QA concluído
```

Este SOP não cria uma suíte automatizada universal.

Ele organiza o QA praticado hoje em uma sequência reproduzível.

2. Gatilho — quando usar

Usar quando:

- as páginas públicas estão suficientemente desenvolvidas;
- a Home e as páginas internas já passaram pela fase principal de construção;
- módulos aplicáveis do Admin estão configurados ou suficientemente maduros para teste;
- o projeto está entrando na revisão interna;
- uma rodada ampla de alterações exige revalidação.

Também pode ser usado de forma parcial durante o desenvolvimento para validar uma área específica.

3. Quando NÃO usar

Não usar como substituto de:

- construção visual;
- debugging detalhado;
- auditoria SEO/AEO/GEO;
- auditoria de segurança;

- publicação;
- smoke de produção específico;
- entrega do Admin.

Procedimentos relacionados

- bug identificado → SOP-009;
 - SEO/AEO/GEO → SOP-010;
 - feedback/ajustes → SOP-011;
 - publicação/entrega → SOP-012.
-

4. Papéis

4.1 Desenvolvedor

Responsável por:

- executar validação manual;
- definir o percurso de teste;
- julgar qualidade visual;
- reproduzir problemas;
- corrigir ou encaminhar bugs;
- decidir quando a rodada está suficientemente validada.

4.2 Codex

Pode ser usado para:

- rodada ampla de testes;
- navegação automatizada;
- build/typecheck quando aplicável;
- inspeção de erros;
- comparação;
- validação dirigida;
- regressão.

4.3 Lovable

Pode apoiar em:

- bugs funcionais;
- backend;
- banco;
- autenticação;
- Supabase;
- correções específicas.

4.4 Claude

Pode apoiar em:

- problemas visuais;
- refinamento;
- responsividade;
- efeitos.

4.5 WSS / revisão humana

Pode realizar segunda revisão interna.

A revisão humana não é substituída pela automação.

5. Pré-condições

Antes de iniciar:

1. projeto acessível;
2. versão atual conhecida;
3. páginas principais implementadas;
4. Sitemap disponível para conferir cobertura;
5. módulos aplicáveis identificados;
6. dados fictícios de teste disponíveis;
7. perfis de teste disponíveis quando aplicável;
8. ambiente suficientemente estável para navegação.

Se a versão estiver desatualizada

Atualizar antes do QA.

Não validar uma versão sabidamente superada.

6. Inputs

Conforme o projeto:

- Sitemap;
- Contexto de Desenvolvimento;
- Home e páginas internas;
- `design.md`;
- referências visuais;
- lista de módulos aplicáveis;
- perfis de teste;

- dados fictícios;
 - `ajustes.md`, se a rodada for pós-feedback;
 - resultado de rodada anterior;
 - erros conhecidos.
-

7. Ferramentas e ambientes

Podem ser usados:

- navegador;
- DevTools;
- desktop/monitor;
- notebook;
- mobile físico;
- Codex;
- Lovable;
- Claude;
- Admin;
- ambiente local;
- preview publicado quando aplicável.

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Não existe matriz universal de:

- navegadores;
- sistemas operacionais;
- resoluções;
- dispositivos.

O AS-IS prioriza cenários reais de uso do Dev.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE I — PREPARAR A RODADA

8. Passo 1 — Confirmar versão e ambiente

Antes de testar:

- confirmar que o site é a versão que deveria ser validada;
- confirmar que não há alteração pendente conhecida;
- confirmar ambiente correto.

Objetivo

Evitar falso bug ou validação inválida por ambiente incorreto.

9. Passo 2 — Definir escopo do QA

Classificar a rodada como:

QA completo

Quando:

- site está entrando em revisão interna;
- houve muitas alterações;
- publicação se aproxima.

QA dirigido

Quando:

- mudança localizada;
- correção específica;
- módulo isolado;
- regressão pós-fix.

Regra

O escopo da validação deve acompanhar o risco da alteração.

10. Passo 3 — Revisar Sitemap e módulos aplicáveis

Antes de navegar, lembrar:

- quais páginas deveriam existir;
- quais funcionalidades se aplicam;

- quais módulos são opcionais.

Não fazer

Testar ausência de módulo não contratado como erro.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE II — QA VISUAL

11. Passo 4 — Percorrer página por página

Percorrer o site público:

```
Home
→ páginas internas
→ Blog quando aplicável
→ demais rotas públicas
```

Em cada página:

- iniciar no Hero;
- seguir seção por seção;
- terminar no Footer.

12. Passo 5 — Observar fidelidade visual

Verificar:

- direção visual;
- consistência;
- tipografia;
- paleta;
- proporções;
- imagens;
- ritmo;
- cards;
- CTAs;
- comportamento.

Importante

O QA não exige que toda página seja idêntica à Home.

Exige coerência com o sistema visual do projeto.

13. Passo 6 — Verificar tipografia

Observar:

- fonte correta;

- peso;
 - hierarquia;
 - tamanho;
 - quebra de linha;
 - legibilidade;
 - headings;
 - textos excessivamente comprimidos/longos.
-

14. Passo 7 — Verificar espaçamentos e alinhamentos

Observar:

- margens;
 - paddings;
 - gaps;
 - alinhamento;
 - largura de conteúdo;
 - consistência entre seções.
-

15. Passo 8 — Verificar imagens e assets

Conferir:

- imagem correta;
 - crop;
 - foco;
 - proporção;
 - qualidade;
 - ausência de stretching;
 - ausência de imagem demo indevida.
-

16. Passo 9 — Verificar efeitos/animações

Observar:

- entrada;
- scroll;
- hover;
- transição;
- parallax;

- motion;
- efeito 3D quando aplicável.

Falhas típicas

- trava;
 - atraso excessivo;
 - pisca;
 - elemento some;
 - sobreposição;
 - scroll horizontal;
 - efeito inadequado em mobile.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE III — QA RESPONSIVO

17. Três cenários principais do AS-IS

O Dev prioriza:

1. **Desktop** — MacBook em monitor;
 2. **Notebook** — MacBook;
 3. **Mobile físico** — aparelho real via rede/local quando aplicável.
-

18. Passo 10 — Validar desktop

Percorrer:

- Home;
- páginas principais;
- Header;
- menus;
- CTAs;
- formulários;
- efeitos;
- Footer.

Observar:

- largura;
 - densidade;
 - alinhamentos;
 - proporções;
 - comportamento de hover/motion.
-

19. Passo 11 — Validar notebook

Repetir percurso relevante.

Objetivo:

- detectar layout que funciona em monitor maior mas quebra em largura menor;
 - observar textos;
 - cards;
 - imagem;
 - menus.
-

20. Passo 12 — Validar mobile físico

Abrir em aparelho real.

Observar:

- menu;
- Hero;
- imagens;
- textos;
- botões;
- formulários;
- Chat/CTA;
- efeitos;
- scroll;
- toque;
- Footer.

Atenção

O mobile pode usar solução diferente do desktop.

Não exigir replicação literal de efeitos.

21. Passo 13 — Procurar scroll horizontal

Em mobile:

- percorrer a página;
- observar elementos fora da viewport;
- testar componentes com movimento;
- conferir carrosséis/cards.

Scroll horizontal inesperado deve ser tratado.

22. AUSÊNCIA ATUAL DE PADRONIZAÇÃO — viewports intermediários

Não existe hoje uma matriz grande e obrigatória de breakpoints testados manualmente em todo projeto.

Consequência

Não documentar uma matriz fictícia.

Viewports adicionais podem ser testados quando:

- existe suspeita;
 - componente é sensível;
 - alteração afeta breakpoint;
 - QA dirigido exige.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IV — NAVEGAÇÃO E ROTAS

23. Passo 14 — Testar navegação principal

Verificar:

- logo;
 - navbar;
 - menu mobile;
 - links do Header;
 - Footer;
 - links contextuais.
-

24. Passo 15 — Testar CTAs

Clicar nos CTAs principais.

Confirmar:

- destino;
- rota;
- ação;
- comportamento.

Caso real Daniel P

CTA apontava para:

```
/sobre-mim
```

mas a rota existente era:

```
/sobre .
```

O resultado era 404.

Lição

Todo CTA relevante precisa ser clicado.

25. Passo 16 — Testar rotas públicas principais

Abrir diretamente páginas importantes.

Verificar:

- renderização;
- conteúdo;

- navegação;
 - ausência de erro.
-

26. 404

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

O AS-IS não possui uma rotina sistemática universal de testar toda classe de 404 em todo projeto.

Aplicação

Testar quando:

- rota foi alterada;
- auditoria apontou problema;
- link foi corrigido;
- projeto possui comportamento crítico de rota.

Normas técnicas podem exigir 404 real em páginas públicas.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE V — QA DE CTAs, FORMULÁRIOS E LEADS

27. Passo 17 — Identificar fluxos de captação

Mapear:

- formulário;
 - CTA;
 - agendamento;
 - Chat;
 - WhatsApp;
 - outras entradas de Lead.
-

28. Passo 18 — Executar formulário ponta a ponta

Fluxo:

1. abrir;
 2. preencher com dados fictícios;
 3. enviar;
 4. observar mensagem de sucesso/erro;
 5. abrir Admin;
 6. localizar Lead;
 7. conferir dados.
-

29. Passo 19 — Conferir Lead

Verificar:

- nome;
- origem;
- interesse;
- especialidade/opção quando aplicável;
- trajetória/source quando disponível.

Critério

Sucesso visual sem persistência no Admin não é suficiente.

30. Passo 20 — Testar erros evidentes do formulário

Quando aplicável:

- campo obrigatório vazio;
- telefone inválido;
- envio incompleto.

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Não existe matriz universal e exaustiva de validação negativa para todo formulário.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VI — QA DE CHAT / CTA / WHATSAPP

31. Passo 21 — Testar Chat

Executar conversa.

Confirmar:

- widget abre;
 - interação funciona;
 - dados são coletados quando aplicável;
 - conversa aparece no Admin.
-

32. Passo 22 — Testar WhatsApp

Quando aplicável:

- clicar no destino;
- conferir número;
- conferir mensagem;
- confirmar canal correto.

Se não se aplica:

- ausência não é erro.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VII — QA DE CHATBOT

33. Quando Chatbot se aplica

O QA detalhado de configuração pertence ao SOP-006.

Neste SOP, revalidar apenas que a capacidade continua funcionando dentro do site completo.

34. Passo 23 — Revalidar fluxo básico do Chatbot

Verificar:

- abre;
 - captura dados;
 - inicia conversa;
 - responde;
 - conversa aparece no Admin;
 - Lead aparece quando obrigatório;
 - handoff continua possível.
-

35. Teste dirigido adicional

Se houve alteração em Chatbot/IA/Base de Conhecimento:

- usar SOP-006;
 - executar benchmark quando aplicável;
 - aumentar a profundidade do QA.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VIII — QA DO ADMIN

36. Passo 24 — Abrir Admin

Confirmar:

- login;
 - carregamento;
 - navegação;
 - módulos aplicáveis.
-

37. Passo 25 — Percorrer módulo por módulo

Conforme projeto:

- Leads;
- Blog;
- Chat;
- Chatbot;
- FAQ;
- depoimentos;
- configurações;
- Search;
- Performance;
- outros.

Não testar módulo não aplicável como falha.

38. Passo 26 — Executar função essencial de cada módulo

Exemplos:

Leads

- abrir;
- buscar;
- conferir registro de teste.

Blog

- abrir gestão;
- conferir conteúdo/configuração;

- validar reflexo público quando necessário.

Chat

- localizar conversa;
- assumir/encerrar quando aplicável.

Configurações

- conferir dados relevantes.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IX — QA DE PERFIS E ACESSO

39. Passo 27 — Testar perfil Equipe/cliente

Confirmar funcionalmente:

- acessa o que deve;
 - não vê superfície Webstar-only evidente;
 - opera módulos previstos.
-

40. Passo 28 — Testar perfil Webstar

Confirmar:

- acesso ampliado esperado;
 - módulos administrativos apropriados;
 - funções Webstar-only quando aplicáveis.
-

41. Limite

Este SOP testa o comportamento funcional visível.

Não substitui auditoria de:

- RLS;
- server-side authorization;
- privilege escalation;
- secrets.

Esses temas pertencem ao Manual 6 e às validações técnicas aplicáveis.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE X — DADOS DE TESTE

42. Passo 29 — Usar dados claramente fictícios

Durante QA:

- Leads;
- conversas;
- posts;
- outros registros;

devem ser identificáveis como teste.

43. Não limpar cedo demais

Dados de teste podem ser úteis durante:

- QA;
- correção;
- regressão.

A limpeza final pertence ao SOP-012.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XI — ERROS, EMPTY STATES E CASOS NÃO COBERTOS

44. Empty states

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Não existe teste sistemático universal de todos os empty states.

Testar quando

- módulo depende fortemente de estado vazio;
 - alteração tocou essa área;
 - problema foi reportado.
-

45. Estados de erro

Não existe matriz exaustiva universal.

Testar de forma dirigida quando:

- fluxo sofreu alteração;
 - existe risco;
 - bug anterior;
 - integração externa relevante.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XII — USO DO CODEX NO QA

46. Passo 30 — Solicitar rodada ampla ao Codex quando útil

O Dev pode pedir ao Codex:

- testar páginas;
 - navegar;
 - clicar;
 - identificar erros;
 - verificar build;
 - executar typecheck;
 - inspecionar console;
 - relatar inconsistências.
-

47. Não delegar decisão final

O relatório do Codex é input.

O Dev ainda precisa:

- interpretar;
 - reproduzir;
 - confirmar;
 - priorizar.
-

48. Build e typecheck

Podem ser executados quando:

- a alteração é técnica;
- o projeto usa esses checks;
- há risco relevante.

AUSÊNCIA

Não são gate universal obrigatório para todo QA.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIII — REGISTRAR PROBLEMAS

49. Passo 31 — Registrar cada falha reproduzível

Para cada problema:

- página/módulo;
 - ação;
 - sintoma;
 - ambiente;
 - dispositivo;
 - evidência;
 - impacto.
-

50. Classificar por camada

Quando possível:

- visual;
- responsivo;
- navegação;
- frontend;
- backend;
- banco;
- autenticação;
- configuração;
- conteúdo.

Objetivo

Facilitar encaminhamento ao SOP-009.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIV — CORREÇÃO E REVALIDAÇÃO

51. Passo 32 — Corrigir problemas localizados

Quando a causa é simples/óbvia e a correção segura:

- corrigir;
 - reexecutar o mesmo teste.
-

52. Passo 33 — Encaminhar problema não trivial

Se exige investigação:

usar SOP-009 — Diagnosticar e Corrigir Bug por Camada.

53. Passo 34 — Revalidar exatamente o fluxo que falhou

Depois da correção:

- repetir ação;
 - confirmar desaparecimento do sintoma;
 - confirmar resultado esperado.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XV — REGRESSÃO

54. Passo 35 — Revalidar áreas impactadas

Depois de corrigir ou alterar:

- testar área corrigida;
 - testar componentes compartilhados relacionados;
 - testar fluxo conectado.
-

55. Regressão visual

Quando alteração visual:

- desktop;
 - notebook;
 - mobile;
 - áreas que reutilizam componente.
-

56. Regressão funcional

Quando alteração funcional:

- ação principal;
 - persistência;
 - Admin;
 - integrações relacionadas.
-

57. Regressão de perfil

Quando alteração de autorização/configuração:

- Webstar;
- Equipe;
- usuário sem papel quando aplicável.

Auditoria profunda pertence à camada de segurança.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVI — REVISÃO HUMANA ADICIONAL

58. Revisão adicional em paralelo

O fluxo atual reconhece valor em outras pessoas revisarem o site a partir do momento em que o Dev sinaliza uma versão ou área revisável. Sempre que possível, essa revisão acontece **em paralelo** à própria revisão do Dev, e não apenas depois de ele encerrar toda a sua validação.

Razão

Quem passa muitas horas na mesma interface pode deixar detalhes passarem; perspectivas simultâneas ajudam a encontrar problemas mais cedo.

Limite

Não existe cadeia universal rígida de revisores formalizada neste SOP.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVII — PROCESSO SITES WSS / MARCO 8

59. Relação com o Marco 8

No Roadmap Sites WSS, o Marco 8 — Revisão Geral e Homologação inclui:

- revisão visual;
- revisão responsiva;
- revisão de textos;
- links;
- pontos de captação;
- Admin;
- Blog;
- SEO básico;
- ajustes finais;
- revisão do cliente quando aplicável.

Este SOP fornece o método técnico de QA que sustenta parte dessa homologação.

60. Processo ≠ SOP

O Marco 8 organiza:

o que precisa estar validado.

O SOP-008 organiza:

como o Dev executa o QA visual, responsivo e funcional.

Não substituir:

- checklist do Flow;
 - evidências formais;
 - aprovação do cliente;
 - decisão de homologação.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVIII — STOP CONDITION

61. Condição de parada

O SOP está concluído quando:

as páginas relevantes foram percorridas nos cenários principais de desktop, notebook e mobile; navegação, CTAs e fluxos essenciais foram testados; módulos operacionais aplicáveis foram verificados; problemas identificados foram corrigidos ou explicitamente encaminhados; e as áreas afetadas passaram por revalidação/regressão proporcional.

62. Checklist mínimo

Antes de encerrar:

- ☐ versão/ambiente corretos;
- ☐ páginas principais percorridas;
- ☐ visual revisado;
- ☐ desktop validado;
- ☐ notebook validado;
- ☐ mobile físico validado;
- ☐ navegação testada;
- ☐ CTAs principais testados;
- ☐ formulários/Leads testados;
- ☐ Chat/WhatsApp testados quando aplicáveis;
- ☐ Chatbot revalidado quando aplicável;
- ☐ Admin percorrido;
- ☐ módulos aplicáveis verificados;
- ☐ perfis testados funcionalmente quando aplicável;
- ☐ problemas registrados;
- ☐ bugs complexos encaminhados ao SOP-009;
- ☐ correções revalidadas;
- ☐ regressão dirigida executada.

63. O que NÃO é necessário para encerrar

Não exigir universalmente:

- suíte automatizada completa;
- cobertura percentual;
- matriz ampla de viewports;

- teste de todo empty state;
 - teste de toda falha externa;
 - auditoria profunda de segurança;
 - auditoria completa de SEO/AEO/GEO;
 - publicação final.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIX — CONTINGÊNCIAS

64. Visual correto em desktop e quebrado em mobile

Corrigir a versão mobile sem alterar desktop validado, quando possível.

Depois:

- revalidar mobile;
 - revalidar desktop para regressão.
-

65. CTA termina em 404

1. reproduzir;
 2. conferir URL;
 3. comparar com rota real;
 4. corrigir path;
 5. testar novamente.
-

66. Formulário mostra sucesso mas Lead não aparece

Considerar fluxo incompleto.

Encaminhar ao SOP-009.

67. Chat abre mas conversa não chega ao Admin

Considerar falha funcional.

Não aprovar a capacidade.

68. Admin abre para Webstar, mas não para Equipe

Verificar se:

- comportamento é esperado;
- capability é restrita;
- existe erro real.

Não ampliar permissão automaticamente.

69. Efeito quebra somente em mobile

Priorizar adaptar; se necessário, simplificar ou substituir. Remover apenas quando a solução continuar inviável, instável ou prejudicial.

Não forçar efeito desktop em mobile.

70. Codex aponta problema que o Dev não reproduz

Não alterar automaticamente.

Reproduzir/confirmar estado atual antes.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XX — AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO

71. Não inventar

O AS-IS não possui:

- suíte automatizada universal;
 - cobertura mínima obrigatória;
 - matriz fixa de browsers;
 - matriz fixa de viewports;
 - teste sistemático universal de empty states;
 - teste universal de todos os 404;
 - checklist universal de erro por formulário;
 - build/typecheck obrigatório em toda rodada;
 - gate numérico de performance;
 - auditoria WCAG formal obrigatória;
 - aprovação automática pelo Codex.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XXI — NORMA TÉCNICA VIGENTE

72. Normas complementam o QA

Quando norma vigente exigir, por exemplo:

- SSR;
- HTML inicial;
- 404 real;
- metadados;
- segurança;
- noindex;
- acessibilidade específica;

essas verificações devem ser aplicadas na camada correta.

Para SEO/AEO/GEO, a autoridade vigente é:

- WSS_Padrao_Tecnico_Operacional_SEO_AEO_GEO_v0.2_FINAL .

O recorte técnico pré-publicação possui checklist normativo próprio:

- WSS_SEO_AEO_GEO_Checklist_PrePublicacao_v0.3 .

A ausência AS-IS de teste sistemático de toda classe de 404 não torna **404 real** opcional onde a norma o exige.

Este SOP não incorpora nem substitui o checklist SEO/AEO/GEO completo.

Importante

Não reclassificar uma norma atual como prática histórica do Dev.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XXII — ANTI-PATTERNS

73. Validar só a Home

Páginas internas podem conter falhas próprias.

74. Validar só desktop

Mobile real é parte do AS-IS.

75. Validar formulário apenas visualmente

É necessário conferir persistência no Admin.

76. Validar Chat apenas pelo widget

É necessário conferir conversa/Admin.

77. Confiar em primeira saída do Codex

Resultado automatizado deve ser interpretado.

78. Corrigir sem revalidar

Fix não é conclusão.

79. Testar só área corrigida quando componente é compartilhado

Pode haver regressão em outras páginas.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XXIII — OUTPUT

80. Saída esperada

Ao final:

Site público

- páginas navegadas;
- visual coerente;
- principais cenários responsivos validados;
- CTAs/links funcionais.

Captação

- formulários testados;
- Leads chegando ao Admin;
- Chat/WhatsApp funcionando quando aplicáveis.

Operação

- Admin acessível;
- módulos aplicáveis testados;
- perfis verificados funcionalmente.

Controle

- problemas corrigidos ou encaminhados;
 - regressão dirigida concluída;
 - projeto pronto para próxima camada de validação.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XXIV — PRÓXIMO PASSO

81. Encaminhamento

Depois do SOP-008:

- bugs complexos → **SOP-009 — Diagnosticar e Corrigir Bug por Camada;**
 - SEO/AEO/GEO → **SOP-010 — Tratar Auditoria SEO, AEO e GEO;**
 - feedback → **SOP-011 — Processar Rodada de Feedback via `ajustes.md`;**
 - publicação → **SOP-012 — Preparar, Publicar, Validar Produção e Entregar Admin.**
-

82. Referências

Manual proprietário

- `WSS_Manual_06_Testes_QA_Debugging_Seguranca_SEO_v1.1_FINAL.md`

Manuais relacionados

- `WSS_Manual_03_Construcao_Visual_v1.1_FINAL.md`;
- `WSS_Manual_04_Projeto_Base_Admin_Funcionalidades_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`.

Evidências técnicas complementares

- `WSS_VAL_004_Troubleshooting_Estudos_de_Caso_v0.1.md`;
- `WSS_VAL_003_Auditorias_SEO_AEO_GEO_Seguranca_v0.1.md`.

Contexto de Processo

- `Roadmap_Sites_WSS_Modelo_Padrao_v0.2_FINAL.md`.
-

Status: FINAL — APROVADO