

PROCEDIMENTO OPERACIONAL

MÉTODO DE DESENVOLVIMENTO DE SITES

SOP-004

Construir Páginas Internas Mantendo Consistência
Visual

VERSÃO

v1.1 FINAL

DATA

11/09/2026

STATUS

FINAL — APROVADO

PROCEDIMENTO OPERACIONAL

SOP-004

Construir Páginas Internas Mantendo Consistência Visual

VERSÃO	v1.1 FINAL
DATA	11/09/2026
STATUS	FINAL — APROVADO
MANUAL PROPRIETÁRIO	Manual 3 — Construção Visual
MANUAIS RELACIONADOS	Manual 2 — Entrada e Contexto do Projeto; Manual 5 — Desenvolvimento com Codex, Lovable e GitHub; Manual 6 — Testes, QA, Debugging, Segurança e SEO
ESCOPO	AS-IS — desenvolvimento das páginas internas após estabilização da linguagem visual da Home

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 — DEFINIR QUAL PÁGINA DESENVOLVER	10
7. Passo 1 — Consultar o Sitemap	11
8. Passo 2 — Priorizar a próxima página	11
9. Passo 3 — Permitir paralelismo quando útil	12
PARTE II — PREPARAR A PÁGINA	13
10. Passo 4 — Identificar a finalidade da página	14
11. Passo 5 — Confirmar as seções previstas	14
12. Passo 6 — Identificar conteúdo disponível e pendente	14
PARTE III — USAR A HOME COMO REFERÊNCIA INTERNA	16
13. Passo 7 — Ler a Home como sistema visual	17
14. Passo 8 — Identificar o que deve permanecer	17
15. Passo 9 — Identificar o que pode variar	18
PARTE IV — PRIMEIRA GERAÇÃO	19
16. Passo 10 — Gerar a primeira versão	20
17. Passo 11 — Limitar o escopo	20
18. Passo 12 — Avaliar a primeira versão	20
PARTE V — REUSO E COMPONENTIZAÇÃO	21
19. Princípio AS-IS	22
20. Passo 13 — Verificar se já existe solução reutilizável	22
21. Passo 14 — Reutilizar quando fizer sentido	22

22. Passo 15 — Não generalizar solução específica sem necessidade	22
23. Passo 16 — Preservar o Design System híbrido do projeto	23
PARTE VI — REFINAMENTO VISUAL	24
24. Passo 17 — Refinar seção por seção	25
25. Passo 18 — Comparar com a Home	25
26. Passo 19 — Evitar cópia literal da Home	25
27. Passo 20 — Ajustar o peso visual à importância da página	26
PARTE VII — REFERÊNCIA ESPECÍFICA E EFEITOS	27
28. Quando uma página possui referência própria	28
29. Subfluxo de efeito	28
30. Regra de contenção	28
PARTE VIII — ASSETS	29
31. Passo 21 — Aplicar imagens disponíveis	30
32. Passo 22 — Validar enquadramento por breakpoint	30
33. Assets ausentes	30
34. AUSÊNCIA ATUAL DE PADRONIZAÇÃO — pipeline de assets	30
PARTE IX — CONTEÚDO	32
35. Passo 23 — Aplicar ou adaptar conteúdo	33
36. Passo 24 — Preservar fatos sensíveis	33
37. Conteúdo provisório	33
PARTE X — CTAS, NAVEGAÇÃO E CONEXÕES ENTRE PÁGINAS	34
38. Passo 25 — Aplicar CTAs relevantes	35
39. Passo 26 — Conferir navegação principal da página	35
40. Caso real de atenção	35
PARTE XI — RESPONSABILIDADE	36
41. Passo 27 — Revisar desktop	37
42. Passo 28 — Revisar notebook	37
43. Passo 29 — Revisar mobile	37
44. Passo 30 — Adaptar quando necessário	37
PARTE XII — PERFORMANCE E ACESSIBILIDADE NO NÍVEL DE CONSTRUÇÃO	39
45. Passo 31 — Observar custo percebido	40
46. Passo 32 — Corrigir problemas óbvios de acessibilidade	40

PARTE XIII — VALIDAÇÃO DA PÁGINA	41
47. Três verificações AS-IS principais	42
48. Verificação adicional de consistência	42
49. Não confundir página pronta com homologação	42
PARTE XIV — CICLO ENTRE PÁGINAS	43
50. Passo 33 — Registrar pendências relevantes	44
51. Passo 34 — Preservar componentes/padrões úteis	44
52. Passo 35 — Seguir para a próxima página	44
PARTE XV — COBERTURA DO SITEMAP	45
53. Quando o conjunto de páginas internas pode avançar	46
54. Relação com o Marco 6 do Processo Sites WSS	46
PARTE XVI — STOP CONDITION	47
55. Condição de parada por página	48
56. Condição de parada do SOP	48
57. O que NÃO é necessário para encerrar	48
PARTE XVII — CONTINGÊNCIAS	49
58. A página ficou igual demais à Home	50
59. A página ficou desconectada da Home	50
60. A página ficou simples demais	50
61. Falta conteúdo	50
62. Falta imagem	50
63. A IA altera outra página/Home	51
64. Um componente compartilhado quebra outra página	51
PARTE XVIII — AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO	52
65. Não inventar	53
PARTE XIX — NORMA TÉCNICA VIGENTE	54
66. Preservar normas aplicáveis	55
PARTE XX — ANTI-PATTERNS	56
67. Criar cada página como se fosse um projeto novo	57
68. Repetir exatamente a Home em todas as páginas	57
69. Criar componente genérico para qualquer bloco específico	57
70. Ignorar componente reutilizável já existente	57

71. Trabalhar página inteira com prompt excessivamente amplo	57
72. Aceitar página visualmente bonita com ações quebradas	57
73. Inventar conteúdo para preencher seção	58
PARTE XXI — OUTPUT	59
74. Saída esperada por página	60
75. Saída esperada do conjunto	60
PARTE XXII — PRÓXIMO PASSO	61
76. Encaminhamento	62
77. Referências	62

1. Finalidade

Transformar a linguagem visual já consolidada na Home em páginas internas:

- coerentes;
- navegáveis;
- visualmente compatíveis com o projeto;
- adequadas ao conteúdo específico de cada página;
- reaproveitando componentes quando fizer sentido;
- sem transformar consistência em repetição mecânica.

Fluxo principal:

```
Home estabilizada
→ Sitemap
→ escolher página
→ definir conteúdo/estrutura
→ gerar primeira versão
→ reutilizar o que faz sentido
→ variar o que precisa variar
→ refinar
→ aplicar assets/conteúdo
→ adaptar efeitos quando aplicável
→ revisar responsividade
→ validar página
→ seguir para próxima
```

2. Gatilho — quando usar

Usar quando:

- o SOP-003 foi concluído;
- a Home já estabelece suficientemente a linguagem visual do projeto;
- o Sitemap está disponível;
- existe pelo menos uma página interna a desenvolver.

3. Quando NÃO usar

Não usar:

- antes de a Home estabelecer a direção visual;
- para redesenhar a Home;
- para definir Sitemap;
- para criar funcionalidade nova fora do escopo;

- para configurar Admin;
 - para executar QA completo;
 - para publicar;
 - para portar Base para projeto legado.
-

4. Papéis

4.1 Desenvolvedor

Responsável por:

- priorizar páginas;
- interpretar a Home como referência;
- decidir nível de sofisticação;
- julgar reuso × variação;
- validar qualidade visual;
- preservar o escopo;
- decidir quando a página está pronta para avançar.

4.2 Codex

Pode ser usado para:

- construção direta;
- fidelidade visual;
- investigação;
- implementação pesada;
- reprodução de comportamento;
- reuso/componentização.

4.3 Claude

Pode ser usado para:

- refinamento visual;
- tipografia;
- acabamento;
- efeitos;
- mobile;
- adaptação de interface.

4.4 Lovable

Pode ser usado conforme a tarefa, principalmente quando:

- a alteração faz mais sentido em sua integração;
- há questão funcional/backend;

- pequenos ajustes são convenientes ali.

A divisão entre ferramentas permanece contextual.

5. Pré-condições

Antes de iniciar:

1. SOP-003 concluído;
 2. Home suficientemente estabilizada;
 3. Sitemap validado disponível;
 4. contexto do projeto acessível;
 5. conteúdo disponível ou pendências identificadas;
 6. assets disponíveis ou necessidade de substituição/provisório conhecida;
 7. ambiente atualizado antes da rodada local;
 8. ferramenta utilizada compreende o projeto.
-

6. Inputs

Obrigatórios

- Home vigente;
- Sitemap;
- Contexto de Desenvolvimento;
- Briefing Final quando necessário;
- conteúdo disponível da página.

Quando existirem

- `design.md`;
 - referência visual específica da página;
 - screenshots;
 - assets;
 - fotos;
 - material de cliente;
 - componentes já existentes;
 - layouts compartilhados;
 - pendências de conteúdo.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE I — DEFINIR QUAL PÁGINA DESENVOLVER

7. Passo 1 — Consultar o Sitemap

Usar o Sitemap para responder:

- quais páginas precisam existir;
- qual é a função de cada página;
- quais páginas são prioritárias;
- quais páginas podem ser mais diretas;
- quais páginas merecem maior refinamento.

O Sitemap orienta **cobertura**.

Ele não determina sozinho o nível visual de cada página.

8. Passo 2 — Priorizar a próxima página

No AS-IS, não existe ordem universal rígida.

A sequência predominante observada após a Home é:

```
Home
→ Contato
→ Blog
```

Contato e Blog costumam vir cedo por serem, em geral, páginas mais simples. A ordem pode mudar por prioridade estratégica, conteúdo, referência, complexidade ou dependência técnica.

Outras práticas observadas:

- Sobre tende a receber atenção alta;
- página geral de Especialidades pode receber refinamento relevante;
- páginas individuais podem ser mais diretas;
- página estrategicamente importante pode receber mais atenção.

Critérios práticos de priorização

- complexidade;
- existência de referência;
- ideia visual disponível;
- importância estratégica;
- velocidade de execução;
- dependência de conteúdo.

DECISÃO HUMANA NECESSÁRIA

Não existe algoritmo de priorização.

9. Passo 3 — Permitir paralelismo quando útil

Se surgir uma solução clara para uma seção de outra página:

- ela pode ser trabalhada antes de a página atual terminar.

Limite

Paralelismo não deve provocar:

- perda de contexto;
- abandono de páginas;
- inconsistência;
- escopo fora de controle.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE II — PREPARAR A PÁGINA

10. Passo 4 — Identificar a finalidade da página

Antes de gerar a primeira versão, entender:

- o que o usuário precisa encontrar;
- qual conteúdo precisa estar presente;
- qual ação principal é esperada;
- qual relação ela tem com a Home e outras páginas.

11. Passo 5 — Confirmar as seções previstas

Usar Sitemap/contexto/conteúdo para listar as seções esperadas.

Regra

Não inventar novas seções estratégicas apenas porque “ficariam bonitas”.

Se uma nova seção implicar:

- escopo novo;
- informação nova;
- funcionalidade nova;
- decisão do cliente;

DECISÃO HUMANA NECESSÁRIA.

12. Passo 6 — Identificar conteúdo disponível e pendente

Classificar:

- conteúdo disponível;
- conteúdo a adaptar;
- conteúdo provisório;
- conteúdo sensível;
- conteúdo ainda pendente.

Importante

Uma pendência não bloqueia automaticamente a construção.

Avaliar impacto concreto.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE III — USAR A HOME COMO REFERÊNCIA INTERNA

13. Passo 7 — Ler a Home como sistema visual

Observar:

- tipografia;
 - cores;
 - espaçamentos;
 - largura;
 - botões;
 - cards;
 - bordas;
 - raios;
 - linguagem de imagem;
 - ritmo;
 - motion;
 - Header;
 - Footer;
 - CTA final;
 - padrões recorrentes.
-

14. Passo 8 — Identificar o que deve permanecer

Elementos frequentemente reaproveitados:

- Header;
- Footer;
- CTA final;
- FAQ quando aplicável;
- botões;
- identidade;
- tipografia;
- cores;
- padrões de espaçamento;
- linguagem geral.

Objetivo

Fazer a página parecer parte do mesmo produto.

15. Passo 9 — Identificar o que pode variar

Podem variar:

- Hero;
- estrutura de seções;
- depoimentos;
- composições;
- efeitos;
- densidade;
- grau de sofisticação.

Princípio

consistência visual não significa páginas idênticas.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IV — PRIMEIRA GERAÇÃO

16. Passo 10 — Gerar a primeira versão

A instrução pode seguir a lógica:

```
com base na Home
+ Sitemap
+ contexto
+ conteúdo da página
→ criar página X
```

Se houver referência específica

Adicionar:

- screenshot;
- link;
- seção relevante;
- comportamento desejado.

17. Passo 11 — Limitar o escopo

Especificar:

- página que pode mudar;
- componentes compartilhados que podem ser usados;
- áreas que não devem ser alteradas;
- Home já validada que deve ser preservada;
- Admin/backend fora de escopo quando não relacionados.

18. Passo 12 — Avaliar a primeira versão

Verificar:

- se as seções previstas aparecem;
- se a página parece pertencer ao mesmo site;
- se está simples demais;
- se há excesso de repetição da Home;
- se o conteúdo encaixa bem;
- se a hierarquia faz sentido.

A primeira versão normalmente recebe refinamento.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE V — REUSO E COMPONENTIZAÇÃO

19. Princípio AS-IS

A componentização é pragmática.

Não existe regra formal como:

“se apareceu duas vezes, obrigatoriamente criar componente”.

20. Passo 13 — Verificar se já existe solução reutilizável

Antes de criar algo novo, procurar:

- componente compartilhado;
- padrão existente;
- layout parametrizado;
- seção adaptável;
- token/variável existente.

21. Passo 14 — Reutilizar quando fizer sentido

Reuso pode preservar:

- consistência;
- velocidade;
- manutenção;
- comportamento.

Exemplos reais de projetos WSS demonstram:

- FAQ compartilhado;
- cards parametrizados;
- layouts orientados por dados.

22. Passo 15 — Não generalizar solução específica sem necessidade

Se a seção:

- é única;
- tem comportamento próprio;

- possui composição muito específica;

ela pode permanecer local.

Objetivo

Evitar overengineering.

23. Passo 16 — Preservar o Design System híbrido do projeto

Quando existirem:

- `design.md`;
- tokens;
- variáveis;
- componentes;
- utilitários;

usar como referências complementares.

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Não existe um Design System único formal universal para todos os sites WSS.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VI — REFINAMENTO VISUAL

24. Passo 17 — Refinar seção por seção

Trabalhar:

- Hero;
- seções informativas;
- cards;
- imagens;
- CTAs;
- FAQs;
- provas sociais;
- blocos especiais.

25. Passo 18 — Comparar com a Home

A cada rodada, verificar:

- tipografia;
- proporções;
- espaçamentos;
- linguagem;
- ritmo;
- contraste;
- acabamento.

Pergunta de controle:

esta página pertence visualmente ao mesmo projeto?

26. Passo 19 — Evitar cópia literal da Home

Não repetir mecanicamente:

- mesmos blocos;
- mesma ordem;
- mesmo Hero;
- mesmos efeitos;

se o conteúdo da página pede outra solução.

27. Passo 20 — Ajustar o peso visual à importância da página

Página prioritária pode receber:

- composição mais elaborada;
- refinamento extra;
- referência própria;
- efeito específico.

Página individual simples pode ser mais direta.

DECISÃO HUMANA NECESSÁRIA

Não existe classificação universal de sofisticação.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VII — REFERÊNCIA ESPECÍFICA E EFEITOS

28. Quando uma página possui referência própria

Se existir referência específica:

1. identificar a seção;
2. fornecer screenshot;
3. fornecer link;
4. pedir estudo;
5. limitar escopo;
6. adaptar à linguagem já consolidada da Home.

29. Subfluxo de efeito

Quando houver efeito relevante, usar a mesma lógica do SOP-003:

```
identificar
→ estudar
→ reproduzir isoladamente
→ adaptar
→ validar
→ simplificar/substituir se necessário
→ remover apenas se a solução continuar inviável
```

30. Regra de contenção

Não usar efeito apenas para “deixar todas as páginas sofisticadas”.

O efeito precisa:

- fazer sentido;
- combinar com a página;
- preservar usabilidade;
- funcionar em mobile.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VIII — ASSETS

31. Passo 21 — Aplicar imagens disponíveis

Fontes podem incluir:

- material do cliente;
 - Drive;
 - assets do projeto;
 - imagens selecionadas;
 - imagens contextuais geradas por IA quando necessário.
-

32. Passo 22 — Validar enquadramento por breakpoint

Verificar:

- foco;
- crop;
- proporção;
- posição;
- legibilidade sobre imagem.

Mobile pode precisar de posicionamento próprio.

33. Assets ausentes

Se faltar material:

- registrar pendência;
 - utilizar solução provisória quando suportada;
 - não transformar provisório em definitivo sem validação.
-

34. AUSÊNCIA ATUAL DE PADRONIZAÇÃO — pipeline de assets

Não existe etapa formal universal separada de:

- compressão;
- formato;
- naming.

Não inventar pipeline neste SOP.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IX — CONTEÚDO

35. Passo 23 — Aplicar ou adaptar conteúdo

A IA pode ajudar a:

- criar;
 - organizar;
 - adaptar;
 - resumir;
 - reescrever.
-

36. Passo 24 — Preservar fatos sensíveis

Não inventar:

- formação;
 - títulos;
 - especialidades;
 - procedimentos;
 - resultados;
 - afirmações médicas;
 - fatos sensíveis equivalentes.
-

37. Conteúdo provisório

Quando aplicável:

- marcar como provisório;
 - manter pendência visível;
 - substituir quando o conteúdo final chegar.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE X — CTAs, NAVEGAÇÃO E CONEXÕES ENTRE PÁGINAS

38. Passo 25 — Aplicar CTAs relevantes

Verificar se a página possui:

- ação principal;
- CTA coerente;
- destino adequado.

39. Passo 26 — Conferir navegação principal da página

No nível de construção, testar:

- botões principais;
- links principais;
- CTA;
- retorno/navegação esperada.

Limite

QA completo pertence ao SOP-008.

40. Caso real de atenção

Um projeto visualmente sofisticado apresentou CTA apontando para rota inexistente.

Lição:

qualidade visual não substitui conferência de ação principal.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XI — RESPONSABILIDADE

41. Passo 27 — Revisar desktop

Observar:

- estrutura;
 - largura;
 - espaçamento;
 - imagens;
 - efeitos;
 - CTA.
-

42. Passo 28 — Revisar notebook

Notebook pode revelar problemas que não aparecem no monitor maior:

- proporções;
 - bordas;
 - overflow;
 - efeitos.
-

43. Passo 29 — Revisar mobile

Verificar:

- leitura;
 - stacking;
 - espaçamento;
 - imagens;
 - CTAs;
 - efeitos;
 - scroll horizontal.
-

44. Passo 30 — Adaptar quando necessário

Mobile pode ter:

- composição diferente;
- efeito simplificado;
- imagem reposicionada;

- elementos removidos.

Não exigir equivalência literal ao desktop.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XII — PERFORMANCE E ACESSIBILIDADE NO NÍVEL DE CONSTRUÇÃO

45. Passo 31 — Observar custo percebido

Se a página fica pesada:

- identificar efeito/asset/componente causador;
- simplificar ou otimizar quando necessário.

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Sem score universal obrigatório nesta etapa.

46. Passo 32 — Corrigir problemas óbvios de acessibilidade

Observar, quando perceptível:

- contraste;
- tamanho de texto;
- nomes acessíveis;
- alt;
- legibilidade.

Limite

Não existe auditoria WCAG formal obrigatória neste SOP.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIII — VALIDAÇÃO DA PÁGINA

47. Três verificações AS-IS principais

Antes de considerar a página pronta na fase de construção:

1. **todas as seções previstas no Sitemap estão presentes;**
2. **a qualidade visual está satisfatória e não parece simples demais;**
3. **botões e ações principais funcionam.**

Essas são as três verificações práticas consolidadas.

48. Verificação adicional de consistência

Confirmar também:

- identidade preservada;
- linguagem compatível com a Home;
- conteúdo coerente;
- ausência de desvio evidente.

Essa verificação não cria um novo checklist formal; apenas explicita o objetivo do SOP.

49. Não confundir página pronta com homologação

"Pronta" nesta etapa não significa:

- QA completo;
- regressão concluída;
- SEO concluído;
- segurança concluída;
- publicação pronta;
- cliente aprovou.

Essas etapas pertencem aos SOPs/Manuais posteriores.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIV — CICLO ENTRE PÁGINAS

50. Passo 33 — Registrar pendências relevantes

Antes de seguir:

- conteúdo faltante;
 - imagem faltante;
 - decisão humana;
 - problema não bloqueante.
-

51. Passo 34 — Preservar componentes/padrões úteis

Se a página criou solução reutilizável que faz sentido para outras:

- mantê-la acessível;
 - reaproveitar posteriormente quando apropriado.
-

52. Passo 35 — Seguir para a próxima página

Voltar ao Sitemap e repetir:

```
priorizar  
→ construir  
→ refinar  
→ validar  
→ registrar pendências  
→ próxima
```

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XV — COBERTURA DO SITEMAP

53. Quando o conjunto de páginas internas pode avançar

O conjunto pode avançar para QA principal quando:

- páginas previstas no recorte de desenvolvimento foram implementadas ou possuem pendência/ justificativa conhecida;
 - navegação principal está utilizável;
 - conteúdo-base está aplicado;
 - páginas estão visualmente coerentes;
 - problemas bloqueantes evidentes foram tratados.
-

54. Relação com o Marco 6 do Processo Sites WSS

No Processo Sites WSS, o Marco 6 trata:

- páginas internas;
- conteúdo-base;
- implementação do Sitemap;
- navegação.

Este SOP documenta **como o Dev constrói essas páginas**.

Ele não substitui:

- checklist formal do Marco;
 - status;
 - evidências;
 - decisão formal de conclusão.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVI — STOP CONDITION

55. Condição de parada por página

Uma página está concluída neste SOP quando:

possui as seções previstas, apresenta qualidade visual satisfatória e coerente com a Home, e suas ações principais funcionam no nível necessário para seguir à próxima etapa.

56. Condição de parada do SOP

O SOP-004 está concluído quando:

as páginas internas necessárias ao estágio atual foram desenvolvidas com coerência visual suficiente para entrar no QA principal, com pendências remanescentes explicitamente registradas.

57. O que NÃO é necessário para encerrar

Não exigir:

- Admin concluído;
- QA completo;
- auditoria SEO;
- auditoria de segurança;
- publicação;
- aprovação final do cliente;
- ausência absoluta de pendências não bloqueantes.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVII — CONTINGÊNCIAS

58. A página ficou igual demais à Home

- revisar finalidade;
 - variar composição;
 - preservar identidade sem repetir estrutura.
-

59. A página ficou desconectada da Home

- reler `design.md`;
 - comparar tokens/padrões;
 - reutilizar elementos recorrentes;
 - revisar tipografia/espaçamento/CTA.
-

60. A página ficou simples demais

- revisar hierarquia;
- composição;
- imagens;
- ritmo;
- referência específica;
- importância estratégica.

Não adicionar complexidade sem função.

61. Falta conteúdo

Avaliar:

- bloqueia a estrutura?
- pode usar provisório?
- precisa de decisão humana?

Registrar a pendência.

62. Falta imagem

Usar solução provisória adequada ou registrar pendência.

Não inventar material factual.

63. A IA altera outra página/Home

1. interromper;
 2. restaurar/preservar área validada;
 3. limitar escopo;
 4. reforçar página específica;
 5. retomar.
-

64. Um componente compartilhado quebra outra página

Tratar como regressão.

Não aceitar a mudança apenas porque melhorou a página atual.

Ver Manual 6 / SOP-008.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XVIII — AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO

65. Não inventar

O AS-IS não possui:

- sequência rígida universal de páginas;
 - score de qualidade visual;
 - regra formal universal de componentização;
 - obrigação de usar template compartilhado;
 - Design System universal formal;
 - número fixo de rodadas;
 - nível obrigatório de sofisticação por tipo de página;
 - matriz universal de breakpoints;
 - score universal de performance;
 - auditoria WCAG obrigatória;
 - pipeline formal universal de assets.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XIX — NORMA TÉCNICA VIGENTE

66. Preservar normas aplicáveis

Quando o projeto estiver sujeito a norma técnica vigente:

- preservar arquitetura pública exigida;
- não sacrificar requisito técnico por conveniência visual;
- manter regras aplicáveis de URL, renderização, indexação e conteúdo;
- tratar a norma como camada própria.

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

- `WSS_Padrao_Tecnico_Operacional_SEO_AEO_GEO_v0.2_FINAL`.

Conteúdo público indexável continua sujeito ao resultado arquitetural exigido por essa norma. Core Web Vitals, `alt` contextual e semântica permanecem requisitos/recomendações da camada normativa quando aplicáveis, sem transformar este SOP visual em procedimento de SEO.

Este SOP não redefine a norma.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XX — ANTI-PATTERNS

67. Criar cada página como se fosse um projeto novo

Isso quebra consistência.

A Home é a referência interna principal.

68. Repetir exatamente a Home em todas as páginas

Consistência não significa duplicação.

69. Criar componente genérico para qualquer bloco específico

Pode gerar overengineering.

70. Ignorar componente reutilizável já existente

Pode aumentar duplicação e divergência.

71. Trabalhar página inteira com prompt excessivamente amplo

Preferir:

- página delimitada;
 - seção;
 - objetivo;
 - referências concretas.
-

72. Aceitar página visualmente bonita com ações quebradas

Botões e ações principais fazem parte do critério prático de pronto.

73. Inventar conteúdo para preencher seção

Especialmente inadequado em fatos sensíveis.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XXI — OUTPUT

74. Saída esperada por página

Cada página deve sair com:

- estrutura prevista implementada;
 - linguagem coerente com a Home;
 - conteúdo-base aplicado;
 - assets aplicados/provisórios identificados;
 - CTAs principais;
 - ações principais funcionais;
 - responsividade suficiente;
 - pendências registradas.
-

75. Saída esperada do conjunto

Ao final do SOP:

- Sitemap materializado no estágio necessário;
 - páginas internas visualmente coerentes;
 - Home preservada como âncora;
 - reuso aplicado de forma pragmática;
 - variações próprias das páginas preservadas;
 - conjunto pronto para QA principal.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XXII — PRÓXIMO PASSO

76. Encaminhamento

Depois do SOP-004, a execução pode seguir em paralelo conforme o projeto para:

- SOP-005 — Configurar Admin e Módulos Aplicáveis;
- SOP-008 — Executar QA Visual, Funcional e Responsivo.

A sequência real pode ter sobreposição, pois o Admin pode continuar em paralelo ao institucional.

77. Referências

Manual proprietário

- WSS_Manual_03_Construcao_Visual_v1.1_FINAL.md

Manuais relacionados

- WSS_Manual_02_Entrada_Contexto_Projeto_v1.1_FINAL.md ;
- WSS_Manual_05_Development_Codex_Lovable_GitHub_v1.1_FINAL.md ;
- WSS_Manual_06_Testes_QA_Debugging_Seguranca_SEO_v1.1_FINAL.md ;
- WSS_Manual_08_Boas_Praticas_AntiPatterns_Depreciadas_Casos_Reais_v1.1_FINAL.md .

SOP anterior

- WSS_SOP_003_Construir_Refinar_Home_Referencias_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 ;
- WSS_Metodo_Development_Sites_Plano_Conducao_v0.3.md .

Evidência técnica complementar

- WSS_VAL_001_DanielP_Componentizacao_DesignSystem_Visual_v0.1.md .
-

Status: FINAL — APROVADO