

WEB STAR STUDIO

MANUAL

MÉTODO DE DESENVOLVIMENTO DE SITES

MANUAL 2
Entrada e Contexto do Projeto

VERSÃO

v1.1 FINAL

DATA

11/09/2026

STATUS

FINAL — APROVADO

MANUAL

MANUAL 2

Entrada e Contexto do Projeto

VERSÃO	v1.1 FINAL
DATA	11/09/2026
STATUS	FINAL — APROVADO
ESCOPO DOCUMENTAL	primeira documentação ampliada — P0 + P1 + P2 relevante
PROJETO	WSS — Método de Desenvolvimento de Sites
FORMALIZAÇÃO	Pacote Corretivo Pós-Entrega v1.1 aprovado pela Coordenação em 11/09/2026; versão canônica vigente, com v1.0 FINAL preservado como baseline histórico.

NAVEGAÇÃO DO DOCUMENTO

Sumário

Estrutura do documento canônico v1.1 FINAL.

1. Objetivo	6
2. Escopo e fronteira	6
3. Posição deste Manual no fluxo WSS	7
PARTE I — HANDOFF WSS → DEV	9
4. Momento de entrada do desenvolvedor	10
5. Contexto de Desenvolvimento Consolidado	10
6. Sitemap	11
7. Briefing Final	11
8. Referências visuais e materiais adicionais	12
PARTE II — HIERARQUIA REAL DAS FONTES	13
9. Três comportamentos coexistentes no AS-IS	14
10. AUSÊNCIA ATUAL DE PADRONIZAÇÃO — hierarquia documental universal	14
11. OPORTUNIDADE FUTURA — FORA DO ESCOPO	15
PARTE III — CONTEXTUALIZAÇÃO INICIAL	16
12. ChatGPT pessoal como primeira camada de organização	17
13. Resultado esperado da contextualização inicial	17
14. Prompt master	18
15. Do prompt master para o Lovable	18
PARTE IV — CONTEXTO PERSISTENTE E CONTEXTO DA TAREFA	19
16. Duas camadas práticas de contexto	20
17. contexto.md	20
18. design.md	21
19. Commits e histórico como apoio de contexto	21
PARTE V — TROCA DE CHAT OU FERRAMENTA	22
20. Entrada em nova sessão	23
21. Codex	23

22. Claude	23
23. Lovable	23
PARTE VI — COMO O DEV ESTRUTURA PROMPTS DEPOIS DO PROMPT MASTER	25
24. Regra prática principal: limitar o escopo	26
25. Criar ou alterar uma seção	26
26. Corrigir um problema visual	26
27. Corrigir um bug funcional	27
28. Investigar antes de editar	27
29. Executar testes	27
30. Comparar com referência	27
PARTE VII — PROTEÇÃO DE ESCOPO	29
31. Preservar áreas já validadas	30
32. Anti-pattern de prompting	30
PARTE VIII — VALIDAÇÃO DE ENTENDIMENTO	31
33. Resumo antes de continuar	32
34. Sinais de que a IA saiu do contexto	32
35. O que o Dev faz quando há desvio	32
36. Indicador de effort/info	33
PARTE IX — PENDÊNCIAS E INFORMAÇÃO INCOMPLETA	34
37. Nem toda pendência bloqueia o início	35
38. Placeholders	35
39. Arquivo .md de pendências	35
40. Informações factuais e sensíveis	35
PARTE X — EXEMPLO REAL DE CONTEXTO DE DESENVOLVIMENTO	37
41. Dr. Guilherme Mendonça	38
PARTE XI — NORMA, AUSÊNCIAS E OPORTUNIDADES	40
42. NORMA TÉCNICA VIGENTE	41
43. AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO	41
44. OPORTUNIDADES FUTURAS — FORA DO ESCOPO	41
PARTE XII — RELAÇÃO COM OS OUTROS MANUAIS E SOP	43
45. Relação com o Manual 3 — Construção Visual	44
46. Relação com o Manual 4 — Projeto-Base, Admin e Funcionalidades	44

47. Relação com o Manual 5 — Codex, Lovable e GitHub	44
48. Relação com o Manual 6 — QA, Debugging, Segurança e SEO	44
49. SOP relacionado	45
50. Critério de conclusão deste Manual	45
51. Fontes principais desta versão	46

1. Objetivo

Este Manual documenta como o projeto passa da WSS para o desenvolvedor e como os materiais recebidos são transformados em **contexto operacional utilizável pelas ferramentas de IA e pelo desenvolvimento**.

Seu foco é a passagem:

```
WSS
→ handoff documental
→ leitura e organização pelo Dev
→ contextualização
→ prompt master
→ contexto persistente no projeto
→ validação de entendimento
→ início técnico do projeto
```

Ao final deste Manual, o leitor deve compreender:

- quais materiais chegam ao Dev;
- como eles são usados hoje;
- qual é a relação real entre Contexto de Desenvolvimento Consolidado, Sitemap e Briefing Final;
- como o Dev organiza o projeto inicialmente no ChatGPT pessoal;
- qual é o papel do prompt master;
- como `contexto.md` e `design.md` funcionam;
- como o contexto é preservado entre Lovable, Codex e Claude;
- como o Dev limita o escopo dos prompts;
- como identifica que uma IA perdeu o contexto ou inventou requisitos;
- como lida com informações ainda pendentes;
- quais decisões e validações continuam humanas.

Este Manual registra o **AS-IS**. Ele não cria um novo modelo de handoff, uma nova hierarquia documental ou um template ideal de prompt.

2. Escopo e fronteira

2.1 O que entra neste Manual

Este Manual cobre:

1. momento de entrada do Dev;
2. materiais recebidos da WSS;
3. Contexto de Desenvolvimento Consolidado;
4. Sitemap;
5. Briefing Final;

6. referências visuais e materiais complementares;
7. relação prática entre essas fontes;
8. contextualização inicial com ChatGPT pessoal;
9. preparação do prompt master;
10. envio de contexto ao Lovable;
11. contexto persistente no projeto;
12. `contexto.md`;
13. `design.md`;
14. reconstrução de contexto em Codex e Claude;
15. escopo de prompts;
16. sinais de perda de contexto;
17. tratamento de requisitos inventados;
18. informações pendentes;
19. placeholders e arquivos `.md`;
20. validação factual e informações sensíveis.

2.2 O que não entra neste Manual

Ficam fora:

- remix da Base Admin em passo a passo;
- criação e conexão do repositório GitHub;
- clone e comandos Git;
- execução técnica de Codex/Lovable/Claude;
- construção detalhada da Home;
- análise detalhada de referências;
- configuração de Admin e módulos;
- QA;
- SEO/AEO/GEO;
- publicação.

Esses assuntos pertencem aos Manuais seguintes.

3. Posição deste Manual no fluxo WSS

Este Manual cobre o intervalo entre:

```
decisões do projeto já consolidadas pela WSS
↓
handoff ao Dev
↓
entendimento operacional
↓
prompt master
↓
contexto suficiente para começar
```

O próximo estágio técnico é tratado principalmente no:

Manual 5 — Desenvolvimento com Codex, Lovable e GitHub

A construção visual decorrente desse contexto pertence ao:

Manual 3 — Construção Visual

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE I — HANDOFF WSS → DEV

4. Momento de entrada do desenvolvedor

No fluxo atual, o Dev entra após receber da WSS o conjunto de informações necessário para iniciar a preparação técnica.

O handoff atual inclui, como núcleo:

1. **Contexto de Desenvolvimento Consolidado;**
2. **Sitemap;**
3. **Briefing Final;**
4. arquivos `.md` com informações do cliente, quando existirem;
5. referências visuais;
6. materiais complementares disponíveis.

A partir desse envio, o Dev assume a preparação técnica e começa a organizar o contexto para as ferramentas que usará no desenvolvimento.

5. Contexto de Desenvolvimento Consolidado

Na intenção operacional atual da WSS, o **Contexto de Desenvolvimento Consolidado** é o principal artefato de entrada para o desenvolvimento.

Sua finalidade é concentrar em uma visão operacional informações que já foram decididas ou organizadas anteriormente no Processo Sites WSS.

Ele pode reunir, conforme o projeto:

- objetivo do site;
- posicionamento;
- público;
- direção visual;
- referências;
- identidade;
- sitemap;
- conteúdos;
- módulos aplicáveis;
- CTAs;
- restrições;
- pendências;
- diretrizes técnicas relacionadas.

O Contexto Consolidado é concebido para reduzir a necessidade de o Dev reconstruir todo o histórico do projeto a partir dos documentos do Flow.

Importante

No AS-IS, o Contexto Consolidado **não é a única fonte consultada pelo Dev.**

Ele recebe e consulta também Sitemap, Briefing Final e demais materiais.

6. Sitemap

O Sitemap chega ao Dev como fonte própria de consulta e validação.

Ele orienta principalmente:

- cobertura de páginas;
- hierarquia pública;
- páginas prioritárias;
- páginas individuais;
- organização geral de navegação;
- escopo estrutural que deverá aparecer no site.

Na construção posterior, o Sitemap continua funcionando como guia de cobertura das páginas internas.

Este Manual não transforma o Sitemap em especificação técnica de rotas. A implementação pertence aos Manuais de construção e desenvolvimento.

7. Briefing Final

O Briefing Final também é enviado ao Dev como fonte de consulta.

Ele sustenta especialmente:

- posicionamento;
- informações do cliente;
- escopo;
- tom;
- objetivos;
- decisões de negócio;
- conteúdos e fatos validados;
- restrições;
- pontos levantados no kickoff.

Na prática atual, o Dev não ignora o Briefing apenas porque existe um Contexto Consolidado.

8. Referências visuais e materiais adicionais

O Dev também pode receber:

- referência visual principal;
- referências secundárias;
- screenshots;
- arquivos `.md`;
- fotos;
- links;
- documentos adicionais;
- conteúdos do cliente;
- outras evidências úteis para o projeto.

A direção visual já tende a chegar consolidada quando o desenvolvimento começa.

O processo de analisar e traduzir essas referências pertence ao Manual 3.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE II — HIERARQUIA REAL DAS FONTES

9. Três comportamentos coexistentes no AS-IS

A descoberta encontrou três formas relacionadas, mas não idênticas, de tratar autoridade documental.

9.1 Intenção operacional da WSS

A WSS concebe o **Contexto de Desenvolvimento Consolidado** como principal artefato de execução.

A intenção é que ele seja suficientemente completo para orientar o desenvolvimento sem exigir que o Dev reconstrua todo o histórico do Processo Sites WSS.

9.2 Prática declarada pelo Dev

Na prática, o Dev carrega e consulta conjuntamente:

- Contexto Consolidado;
- Sitemap;
- Briefing Final;
- arquivos `.md`;
- referências;
- demais materiais recebidos.

Portanto, a execução atual não se baseia em leitura exclusiva de uma única fonte.

9.3 Caso real — Dr. Guilherme Mendonça

O arquivo:

`Dr_Guilherme_Mendonca_Contexto_Developolvimento_v1.0_FINAL.md`

foi criado como documento operacional de entrada, mas registra explicitamente que **não substitui as fontes canônicas do projeto**.

Nesse caso específico, o próprio documento define uma ordem de precedência que começa pelo Briefing Curado Pós-Kickoff, seguido pelo Sitemap Validado e por outros documentos.

Isso demonstra que um Contexto Consolidado pode carregar **uma regra de precedência própria do projeto**.

10. AUSÊNCIA ATUAL DE PADRONIZAÇÃO — hierarquia documental universal

Não existe hoje, no Método de Desenvolvimento documentado, uma regra universal única e formalmente estabilizada dizendo:

Contexto Consolidado sempre vence Sitemap

OU:

Briefing sempre vence Contexto Consolidado

O que está sustentado é:

- a WSS pretende que o Contexto Consolidado seja o principal artefato de entrada;
- o Dev consulta as fontes em conjunto;
- um projeto real pode possuir ordem de precedência explicitamente definida dentro do próprio pacote.

Consequência documental

Este Manual **não escolhe** uma nova hierarquia.

Quando um projeto ou artefato trazer uma ordem de precedência explícita, essa regra deve ser respeitada como parte do contexto daquele projeto.

Quando não houver regra explícita e surgir uma contradição material, a divergência deve ser tratada como decisão humana, não resolvida por invenção da IA.

11. OPORTUNIDADE FUTURA — FORA DO ESCOPO

O baseline de descoberta já registrou como oportunidade futura:

alinhar formalmente a hierarquia entre Contexto Consolidado, Sitemap e Briefing Final.

Este Manual apenas preserva esse achado.

Não é objetivo desta versão desenhar essa hierarquia futura.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE III — CONTEXTUALIZAÇÃO INICIAL

12. ChatGPT pessoal como primeira camada de organização

Antes de criar ou alterar o projeto, o Dev usa normalmente seu **ChatGPT pessoal**.

Nessa etapa ele:

- carrega documentos e referências;
- resume o projeto;
- organiza informações;
- procura compreender o caminho geral;
- avalia por onde começar;
- prepara a instrução inicial do projeto.

O objetivo não é desenvolver o site dentro dessa conversa.

O ChatGPT pessoal funciona como uma camada de **organização e preparação do contexto**.

O Claude no terminal pode ser usado excepcionalmente nessa fase quando o resultado obtido com ChatGPT não satisfaz.

13. Resultado esperado da contextualização inicial

A contextualização inicial deve deixar o Dev com entendimento suficiente sobre:

- quem é o cliente;
- objetivo do projeto;
- posicionamento;
- público;
- sitemap;
- direção visual;
- referências;
- conteúdos relevantes;
- restrições;
- módulos/capacidades aplicáveis;
- pendências;
- riscos ou pontos de atenção;
- caminho inicial de desenvolvimento.

No AS-IS, esse entendimento é consolidado pelo Dev com apoio da IA antes da implementação principal.

14. Prompt master

Dessa etapa nasce o **prompt master**.

No fluxo atual, ele funciona como a instrução inicial que leva o contexto do projeto para o início da implementação, principalmente no Lovable.

O prompt master é preparado depois que o Dev:

- leu os materiais;
- organizou as informações;
- formou uma visão geral;
- decidiu por onde começar.

O que este Manual não deve inventar

A descoberta não consolidou um **template canônico único** de prompt master para todos os projetos.

Portanto, esta documentação não cria uma estrutura fixa artificial.

O que está suficientemente sustentado é sua função:

transformar o entendimento acumulado do projeto em uma instrução inicial suficientemente contextualizada para começar o desenvolvimento.

15. Do prompt master para o Lovable

No fluxo atual, o projeto começa no Lovable.

O Dev normalmente:

1. remixa a Base Admin;
2. envia o prompt master;
3. fornece documentos e referências;
4. usa os primeiros prompts para contextualizar o projeto;
5. gera a primeira base do institucional;
6. chega ao primeiro ou segundo protótipo.

A partir daí o projeto é conectado ao GitHub e segue também para o ambiente local.

O procedimento técnico dessa transição pertence ao Manual 5.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IV — CONTEXTO PERSISTENTE E CONTEXTO DA TAREFA

16. Duas camadas práticas de contexto

No fluxo atual é útil distinguir duas camadas observadas.

16.1 Contexto persistente do projeto

Fica principalmente no próprio projeto e em seus artefatos.

Exemplos:

- código existente;
- `contexto.md` ;
- `design.md` ;
- outros arquivos `.md` ;
- commits;
- histórico disponível;
- documentos do projeto.

Essa camada permite que uma nova sessão reconstrua a história e a lógica geral.

16.2 Contexto específico da tarefa

É reforçado no prompt atual.

Pode incluir:

- tela;
- seção;
- módulo;
- bug;
- breakpoint;
- screenshot;
- link de referência;
- restrições;
- áreas que não devem ser alteradas;
- objetivo específico daquela rodada.

O contexto persistente reduz a necessidade de repetir tudo; o contexto da tarefa limita o escopo da execução atual.

17. `contexto.md`

O Dev usa `contexto.md` como arquivo persistente de consolidação do desenvolvimento.

Ele pode ajudar ferramentas futuras a entender:

- projeto;

- decisões;
- estado;
- contexto acumulado.

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Não existe hoje uma regra de atualizar `contexto.md` em toda troca de ferramenta, a cada prompt ou após cada alteração.

Ele funciona como referência persistente e é reforçado quando necessário.

Este Manual não cria uma cadência obrigatória nova.

18. `design.md`

O `design.md` é uma referência persistente especialmente importante para a direção visual.

O Dev declarou que Codex e Claude usam esse arquivo para preservar, entre páginas e chats:

- tipografia;
- espaçamentos;
- cores;
- referências;
- linguagem visual;
- padrões gerais de design.

Quando a ferramenta começa a sair da direção visual, o Dev reforça novamente o `design.md`.

O detalhamento de como essa direção se materializa no código pertence ao Manual 3.

19. Commits e histórico como apoio de contexto

O Dev também usa o próprio estado do projeto, commits e histórico disponível como sinais de continuidade.

Ao trocar de ferramenta, a IA pode reconstruir parte do contexto a partir de:

- repositório;
- arquivos;
- commits;
- histórico acessível no ambiente.

Neste Manual, esses elementos são tratados apenas como **fontes de contexto**.

A operação Git pertence ao Manual 5.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE V — TROCA DE CHAT OU FERRAMENTA

20. Entrada em nova sessão

Ao iniciar uma nova sessão com Codex ou Claude, o Dev costuma pedir que a ferramenta:

- leia o projeto;
- entenda a estrutura;
- leia ou releia os documentos;
- entenda quem é WSS;
- entenda quem é o Dev;
- entenda quem é o cliente;
- apresente um resumo antes de continuar.

Quando um arquivo não está acessível à ferramenta, ele é enviado novamente.

Essa prática reduz a chance de começar uma alteração com uma compreensão incompleta do projeto.

21. Codex

Depois do clone, o Codex consegue ler o repositório e os documentos disponíveis.

No fluxo do Dev, ele também pode reconstruir contexto a partir do que foi trabalhado anteriormente no projeto e, em algumas situações, operar o Lovable pelo navegador.

O uso operacional detalhado pertence ao Manual 5.

22. Claude

Ao passar do Codex para o Claude, o Dev espera que o Claude reconstrua o contexto por:

- projeto;
- arquivos `.md`;
- commits;
- histórico disponível no ambiente.

O Claude é usado com frequência para refinamento visual, o que torna o `design.md` particularmente relevante nessa transição.

23. Lovable

O Lovable é a ferramenta em que o Dev percebe maior chance de perda de contexto quando retorna depois de trabalhar fora dele.

Mesmo assim, essa perda não foi descrita como problema recorrente ou bloqueante no fluxo atual.

Quando necessário, o Dev reforça novamente:

- documentos;
 - regras;
 - objetivo;
 - contexto do projeto;
 - escopo da tarefa.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VI — COMO O DEV ESTRUTURA PROMPTS DEPOIS DO PROMPT MASTER

24. Regra prática principal: limitar o escopo

O Dev procura cercar a tarefa no próprio prompt.

Em vez de pedir alterações muito amplas, normalmente delimita:

- uma tela;
- uma seção;
- um módulo;
- um problema;
- um breakpoint;
- uma funcionalidade.

Essa limitação é uma das práticas atuais mais importantes para reduzir desvios da IA.

25. Criar ou alterar uma seção

Quando trabalha apenas uma parte da interface, o Dev especifica o recorte exato.

Exemplo de comportamento atual:

- se o foco é mobile, declara que desktop já está validado e não deve mudar;
- se o foco é desktop, evita que a ferramenta altere mobile sem necessidade;
- reforça `design.md` quando a direção visual precisa permanecer intacta.

O conteúdo visual detalhado desses prompts pertence ao Manual 3.

26. Corrigir um problema visual

O Dev tende a fornecer evidência concreta:

- print;
- link;
- referência;
- efeito específico.

Quando existe comportamento complexo, pode pedir à ferramenta que **estude a referência antes de editar**.

A abordagem genérica de apenas mandar um site e pedir “copie com fidelidade” é considerada fraca no AS-IS.

27. Corrigir um bug funcional

O Dev delimita o problema e pede investigação focada.

Para bugs funcionais, usa com frequência:

- Lovable;
- Codex.

Em problemas difíceis, pode comparar as investigações antes de seguir.

O método de debugging pertence ao Manual 6.

28. Investigar antes de editar

Quando o resultado depende de referência, efeito ou solução não trivial, o Dev pode pedir primeiro:

- pesquisa;
- leitura;
- comparação;
- avaliação de alternativa;
- estudo do efeito.

Isso evita transformar a primeira hipótese da IA em implementação imediata.

29. Executar testes

Não há um template rígido de prompt de testes no AS-IS.

O Dev delimita:

- área;
- funcionalidade;
- fluxo;

e procura impedir que a ferramenta expanda o trabalho para partes já aprovadas.

A estratégia completa de testes pertence ao Manual 6.

30. Comparar com referência

A prática visual atual trabalha por partes.

Em vez de pedir fidelidade ao site inteiro de uma vez:

1. escolhe seção;

2. fornece print;
3. fornece link;
4. pede estudo da referência;
5. compara resultado;
6. refina;
7. depois adapta ao projeto.

Esse procedimento pertence principalmente ao Manual 3 e aos SOPs de Home/páginas.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VII — PROTEÇÃO DE ESCOPO

31. Preservar áreas já validadas

O Dev usa instruções explícitas para reduzir alterações colaterais.

Exemplos declarados:

- mobile não deve alterar desktop já validado;
- desktop não deve quebrar mobile validado;
- frontend não deve provocar alteração desnecessária no backend;
- backend não deve gerar mudança de interface sem necessidade;
- trabalho no Admin não deve alterar o institucional;
- um módulo do Admin não deve quebrar módulos já validados.

Formulação AS-IS

A proteção do escopo depende fortemente da **clareza do prompt e da validação humana posterior**.

Não existe uma trava técnica universal que impeça a IA de tocar outra área.

32. Anti-pattern de prompting

Um padrão declarado como problemático é:

fornecer apenas o link de uma referência e pedir para a ferramenta ser “extremamente fiel”.

O problema é o excesso de liberdade de interpretação.

A abordagem que tende a funcionar melhor é:

- reduzir escopo;
- dar evidência concreta;
- pedir estudo;
- trabalhar em etapas;
- preservar áreas já aprovadas.

O catálogo completo de anti-patterns pertence ao Manual 8.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE VII — VALIDAÇÃO DE ENTENDIMENTO

33. Resumo antes de continuar

Uma prática usada pelo Dev ao entrar em nova sessão é pedir que a ferramenta **resuma o projeto antes de executar**.

Esse resumo funciona como verificação prática de que ela:

- leu os arquivos;
- entendeu o cliente;
- reconheceu o contexto;
- sabe a área que será trabalhada.

Não existe um formulário formal universal de aceite do resumo.

A validação é feita pelo julgamento do Dev.

34. Sinais de que a IA saiu do contexto

O Dev identifica desvio quando a ferramenta:

- deixa de seguir o design explicado;
- adiciona textos aleatórios sem relação com o projeto;
- cria requisito ou elemento que nunca foi pedido;
- começa a alterar áreas fora do escopo;
- ignora regras já explicadas;
- quebra uma área previamente validada.

Esses sinais valem mais, no AS-IS, do que uma checagem documental automática formal.

35. O que o Dev faz quando há desvio

Quando percebe perda de contexto ou invenção:

1. interrompe a execução;
2. explica que a ferramenta saiu do contexto;
3. reforça documentos e regras relevantes;
4. reduz novamente o escopo;
5. divide a tarefa em partes menores;
6. avança gradualmente.

Em vez de tentar corrigir todo o desvio em um único prompt, o Dev prefere reconduzir a ferramenta em etapas.

36. Indicador de effort/info

O Dev também declarou observar o indicador de `effort/info` da ferramenta como um sinal prático.

Na sua experiência:

- effort baixo pode coincidir com maior perda de contexto;
- effort mais alto tende a preservar melhor o entendimento.

Limite

Isso é uma **heurística pessoal declarada**, não uma regra formal do Método WSS nem uma relação técnica comprovada.

Não deve ser transformado em threshold obrigatório.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IX — PENDÊNCIAS E INFORMAÇÃO INCOMPLETA

37. Nem toda pendência bloqueia o início

O AS-IS permite que certas informações permaneçam pendentes enquanto o desenvolvimento avança.

Um caso real documentado, Dr. Guilherme Mendonça, instrui que pendências do cliente sejam tratadas como **não bloqueantes**, salvo quando forem indispensáveis para publicar determinado conteúdo factual.

Essa formulação pertence ao caso e não deve ser automaticamente universalizada para todos os projetos.

No método geral, a decisão depende do tipo de pendência e do impacto da informação ausente.

38. Placeholders

Quando falta uma informação operacional simples, o Dev pode trabalhar com placeholder.

Exemplo declarado:

- telefone pendente → número provisório no formato `(81) 9 9999-9999`.

O uso de placeholder não significa que a informação foi validada.

Antes da entrega/publicação aplicável, o dado precisa ser substituído ou tratado conforme o caso.

39. Arquivo `.md` de pendências

Para outros conteúdos ainda não recebidos, o Dev registra pendências em arquivo `.md`.

Esse arquivo funciona como memória operacional.

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

O próprio Dev reconhece que esse controle depende bastante de acompanhamento humano e existe risco de um item ser esquecido.

Não há hoje, no AS-IS, mecanismo formal universal que garanta automaticamente a resolução de toda pendência registrada em `.md`.

40. Informações factuais e sensíveis

O Dev evita inventar ou modificar com IA, sem validação:

- formação;
- títulos;
- procedimentos;

- resultados;
- especialidades;
- afirmações médicas;
- outros fatos sensíveis do cliente.

Quando o conteúdo é criado ou adaptado por IA, a validação factual final fica principalmente com o cliente.

Essa prática não equivale a uma política formal universal de revisão clínica/editorial em todos os projetos.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE X — EXEMPLO REAL DE CONTEXTO DE DESENVOLVIMENTO

41. Dr. Guilherme Mendonça

O arquivo:

`Dr_Guilherme_Mendonca_Contexto_Desenvolvimento_v1.0_FINAL.md`

é um exemplo concreto de documento de entrada para desenvolvimento.

Ele reúne em um único artefato operacional, entre outros elementos:

- fontes canônicas e precedência;
- resumo do projeto;
- público;
- princípios de UX;
- direção visual;
- referências;
- paleta;
- posicionamento;
- linguagem;
- arquitetura pública;
- estrutura das páginas;
- Blog;
- SEO/AEO/GEO;
- animações/performance;
- captação e CTAs;
- temas específicos;
- conteúdo provisório;
- pendências.

41.1 O que este caso demonstra

O caso mostra que um Contexto de Desenvolvimento pode funcionar como **ponte operacional** entre decisões consolidadas e implementação.

Também mostra que o próprio contexto pode carregar:

- limites;
- regras;
- fontes;
- pendências;
- placeholders;
- referências normativas.

41.2 O que não deve ser generalizado

A estrutura exata do arquivo do Dr. Guilherme não deve ser tratada automaticamente como template universal do Método WSS.

O caso é evidência real de um handoff, não definição de um novo padrão documental.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XI — NORMA, AUSÊNCIAS E OPORTUNIDADES

42. NORMA TÉCNICA VIGENTE

Documentos normativos técnicos podem chegar ao desenvolvimento ou ser referenciados pelo Contexto Consolidado.

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

- `WSS_Padrao_Tecnico_Operacional_SEO_AEO_GEO_v0.2_FINAL.md` — **padrão canônico vigente**.

O antigo `WSS_Padrao_Tecnico_Arquitetura_SEO_v0.1.md` permanece apenas como baseline histórico/proveniência, não como autoridade concorrente.

Quando uma norma estiver explicitamente aplicável, ela deve ser preservada durante o desenvolvimento.

Entretanto:

uma norma técnica não deve ser apresentada como descrição do que o Dev sempre praticou historicamente.

A camada normativa permanece distinta do AS-IS.

43. AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO

Este Manual registra como ausências reais:

- não existe hierarquia universal única estabilizada entre Contexto Consolidado, Sitemap e Briefing Final;
- não existe obrigação de atualizar `contexto.md` a cada troca de ferramenta;
- não existe obrigação de atualizar `design.md` a cada troca de ferramenta;
- não existe validação formal automatizada de que a IA “entendeu” o projeto;
- a validação de entendimento depende do resumo e julgamento do Dev;
- não há mecanismo automático universal para impedir alterações fora do escopo;
- controle de pendências em `.md` depende de acompanhamento humano;
- não foi consolidado um template canônico único de prompt master para todos os projetos.

Essas ausências não devem ser silenciosamente preenchidas nesta documentação.

44. OPORTUNIDADES FUTURAS — FORA DO ESCOPO

Já identificadas nas fontes:

- alinhar formalmente a hierarquia entre Contexto Consolidado, Sitemap e Briefing Final;
- eventual evolução do controle de pendências;
- eventual padronização adicional de artefatos de entrada.

Nenhuma dessas oportunidades é desenvolvida neste Manual.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE XII — RELAÇÃO COM OS OUTROS MANUAIS E SOP

45. Relação com o Manual 3 — Construção Visual

O Manual 2 entrega:

- contexto compreendido;
- direção;
- referências;
- restrições;
- `design.md`.

O Manual 3 explica como isso é transformado em Home, páginas, componentes, conteúdo, efeitos e consistência visual.

46. Relação com o Manual 4 — Projeto-Base, Admin e Funcionalidades

O Manual 2 identifica:

- módulos/capacidades aplicáveis;
- restrições do cliente;
- contexto do projeto.

O Manual 4 explica como a Base Admin e suas capacidades são configuradas/adaptadas.

47. Relação com o Manual 5 — Codex, Lovable e GitHub

O Manual 2 explica **como o contexto é criado e preservado**.

O Manual 5 explica **como as ferramentas e o repositório são operados**.

Não se deve duplicar, no Manual 2, clone, Git, branch, commit ou rollback.

48. Relação com o Manual 6 — QA, Debugging, Segurança e SEO

Este Manual trata sinais de que a IA entendeu ou não entendeu o projeto.

O Manual 6 trata a validação do **produto implementado** e a investigação de falhas técnicas.

49. SOP relacionado

Este Manual é a base principal para:

SOP-001 — Contextualizar um Novo Projeto WSS

O SOP formaliza este conhecimento em sequência executável, incluindo:

- gatilho;
- inputs;
- leitura;
- organização;
- prompt master;
- validação de entendimento;
- stop condition.

O SOP já foi produzido e aprovado materialmente; sua formalização versionada ocorre em gate posterior ao fechamento dos Manuais.

50. Critério de conclusão deste Manual

Este Manual cumpre seu objetivo quando o leitor consegue responder:

1. quais documentos chegam ao Dev;
 2. como eles são utilizados na prática;
 3. qual é a diferença entre intenção da WSS e prática do Dev na hierarquia de fontes;
 4. como o ChatGPT pessoal entra no fluxo;
 5. para que serve o prompt master;
 6. o que é contexto persistente;
 7. o papel de `contexto.md`;
 8. o papel de `design.md`;
 9. como o contexto é recuperado ao trocar de ferramenta;
 10. como o Dev limita o escopo dos prompts;
 11. como percebe perda de contexto;
 12. o que faz quando a IA inventa requisitos;
 13. como lida com informações pendentes;
 14. quais fatos não devem ser inventados;
 15. quais pontos continuam sem padronização formal.
-

51. Fontes principais desta versão

Baseline canônico

1. WSS_Arqueologia_Tecnica_ASIS_v0.3.md ;
2. WSS_Matriz_Cobertura_ASIS_P0_P1_P2_v0.3_FINAL.md ;
3. WSS_Handoff_Consolidacao_Manuais_SOPs_Skills_v0.2.md ;
4. WSS_Metodo_Desenvolvimento_Sites_Plano_Conducao_v0.3.md ;
5. WSS_Metodo_Desenvolvimento_Sites_Indice_Mestre_v0.1_DRAFT.md .

Evidência humana consultada para nuance

- questionario_metodo_desenvolvimento_AS-IS_perguntas_1_a_70.docx ;
- respostas p1 p2.txt .

Exemplo real

- Dr_Guilherme_Mendonca_Contexto_Desenvolvimento_v1.0_FINAL.md .

Norma técnica vigente relacionada

- WSS_Padrao_Tecnico_Operacional_SE0_AE0_GEO_v0.2_FINAL.md .

Versões anteriores/superadas de AS-IS, Matriz, Handoff e Plano não foram usadas como baseline ativo.

Status: FINAL — APROVADO