

WEB STAR STUDIO

MANUAL

MÉTODO DE DESENVOLVIMENTO DE SITES

MANUAL 1
Visão Geral e Toolchain

VERSÃO

v1.1 FINAL

DATA

11/09/2026

STATUS

FINAL — APROVADO

MANUAL

MANUAL 1

Visão Geral e Toolchain

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	5
2. Escopo e fronteira	5
3. Como ler esta documentação	6
4. Processo Sites WSS × Método de Desenvolvimento de Sites WSS	7
5. Onde o trabalho do desenvolvedor começa	8
6. Onde o trabalho do desenvolvedor termina	9
7. Entradas principais	9
8. Saídas principais	10
9. Fluxo macro AS-IS	10
10. Papéis no fluxo	12
11. Tipos de projeto relevantes ao método	13
PARTE II — TOOLCHAIN WSS	15
12. Visão geral do toolchain	16
13. ChatGPT pessoal	16
14. Lovable	17
15. GitHub	18
16. Codex	18
17. Claude	19
18. Supabase / Lovable Cloud	19
19. Navegador e DevTools	20
20. Ferramentas de referência visual	21
21. SEOmator	21
22. Flow e ferramentas operacionais adjacentes	22
PARTE III — CRITÉRIOS DE ESCOLHA DAS FERRAMENTAS	23
23. Regra prática atual	24
24. Critérios observados de roteamento	24
25. Trabalho manual do Dev como camada do método	25

PARTE IV — GOVERNANÇA E LIMITES	27
26. Fontes de conhecimento do método	28
27. Decisões humanas que permanecem centrais	28
28. NORMA TÉCNICA VIGENTE	29
29. AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO relevantes para esta visão geral	29
30. Práticas atuais que explicam o desenho do toolchain	29
PARTE V — EXEMPLO REAL E NAVEGAÇÃO DA COLEÇÃO	31
31. Exemplo real resumido — Dr. Guilherme Mendonça	32
32. Relação com os demais Manuais	32
33. SOPs relacionados	33
34. Critério de conclusão deste Manual	33
35. Fontes principais desta versão	34

1. Objetivo

Este Manual apresenta a visão geral do **Método de Desenvolvimento de Sites WSS** e o ecossistema de ferramentas utilizado no fluxo atual.

Sua finalidade é permitir que um novo desenvolvedor, colaborador técnico ou agente de IA compreenda, antes de executar qualquer procedimento:

- o que o Método de Desenvolvimento de Sites WSS documenta;
- onde ele começa e termina;
- como ele se relaciona com o Processo Sites WSS;
- quais são as entradas e saídas principais;
- quem participa do fluxo;
- como o trabalho ocorre de ponta a ponta;
- para que cada ferramenta é usada hoje;
- como as ferramentas se relacionam;
- quais decisões permanecem humanas;
- onde não existe padronização formal no AS-IS.

Este Manual é deliberadamente **orientador**. Ele não substitui os Manuais específicos de entrada/contexto, construção visual, Base Admin, desenvolvimento com Codex/Lovable/GitHub, QA, publicação ou conhecimento acumulado.

2. Escopo e fronteira

2.1 O que entra neste Manual

Este Manual cobre:

1. conceito do Método de Desenvolvimento de Sites WSS;
2. fronteira entre Processo Sites WSS e Método de Desenvolvimento;
3. visão ponta a ponta do fluxo atual;
4. principais entradas e saídas;
5. papéis envolvidos;
6. tipos de projeto relevantes ao método;
7. visão funcional do toolchain:
 - ChatGPT pessoal;
 - Lovable;
 - GitHub;
 - ambiente local;
 - Codex;
 - Claude;

- Supabase / Lovable Cloud;
 - navegador e DevTools;
 - Awwwards e fontes visuais auxiliares;
 - SEOmator;
 - ferramentas complementares;
8. critérios práticos de escolha entre ferramentas;
 9. papel do julgamento humano;
 10. ausências atuais de padronização relevantes para a visão geral.

2.2 O que não entra neste Manual

Ficam deliberadamente fora:

- procedimento detalhado de contextualização e criação do prompt master;
- comandos e rotinas de Git;
- clone, pull, commit e push em passo a passo;
- construção detalhada de Home e páginas internas;
- componentização e Design System em profundidade;
- configuração do Admin e de seus módulos;
- arquitetura detalhada do Chatbot;
- execução de QA;
- debugging;
- auditorias de SEO/AEO/GEO;
- validações de segurança;
- publicação, domínio e DNS em passo a passo;
- catálogo completo de troubleshooting;
- estudos de caso completos.

Esses assuntos pertencem aos demais Manuais e SOPs da coleção.

3. Como ler esta documentação

A documentação do Método WSS separa quatro camadas.

3.1 AS-IS

Descreve o que é efetivamente praticado hoje, declarado pelos responsáveis, observado em projetos ou tecnicamente comprovado.

É a camada principal deste Manual.

3.2 AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Registra situações em que hoje não existe regra, checklist, threshold, ritual ou procedimento formal único.

A ausência é um achado válido. Ela não deve ser silenciosamente preenchida com uma prática ideal.

3.3 NORMA TÉCNICA VIGENTE

Registra exigências aprovadas em documentos normativos próprios.

Uma norma vigente pode complementar o método, mas não deve ser apresentada como se sempre tivesse feito parte do AS-IS do desenvolvedor.

3.4 OPORTUNIDADE FUTURA — FORA DO ESCOPO

Registra, quando necessário, uma possível melhoria ou formalização futura.

Esta camada não deve ser usada para desenhar o TO-BE durante a documentação do método atual.

3.5 Convenções terminológicas da coleção

Para reduzir variações desnecessárias, esta coleção adota as seguintes convenções editoriais:

- **Base Admin WSS** — nome preferencial da base reutilizável; **Base** pode ser usada como forma curta depois de apresentada;
- **Admin** — forma curta de Admin do Site/painel administrativo quando o contexto já estiver claro;
- **Web Star Studio (WSS)** — nome institucional; `admin_webstar` e `admin_equipe` permanecem nomes técnicos quando a implementação exigir precisão;
- **perfil Equipe/cliente** — forma funcional para o perfil operacional do cliente; quando a implementação concreta for relevante, usar `admin_equipe`;
- **Chat/CTA** — capacidade persistente de interação/captação; WhatsApp permanece canal condicional, não sinônimo obrigatório de Chat;
- **Claude** — referência à família/ferramenta no fluxo; **Claude Code** deve ser usado quando a referência for especificamente ao ambiente de código/terminal.

Os nomes técnicos existentes em código, configuração ou interface não devem ser renomeados apenas para uniformização editorial.

4. Processo Sites WSS × Método de Desenvolvimento de Sites WSS

Os dois conceitos são relacionados, mas não são equivalentes.

4.1 Processo Sites WSS

O **Processo Sites WSS** organiza o projeto em termos de:

- marcos;
- condução;
- decisões;
- entregáveis;
- validações;
- pendências;
- acompanhamento interno;
- acompanhamento do cliente.

Ele responde principalmente:

o que precisa acontecer no projeto e em que estágio o projeto se encontra?

4.2 Método de Desenvolvimento de Sites WSS

O **Método de Desenvolvimento de Sites WSS** documenta como o desenvolvedor transforma os insumos já consolidados em um site efetivamente implementado.

Ele responde principalmente:

como o desenvolvedor executa tecnicamente o trabalho, com quais ferramentas, quais decisões, quais verificações e quais caminhos alternativos?

Na prática, o Método de Desenvolvimento se conecta principalmente aos marcos de execução, configuração, homologação e publicação do Processo Sites WSS.

O Processo não deve ser reescrito neste Manual. Quando um conceito de marco, entregável ou condução for necessário, a documentação do Processo Sites WSS deve ser tratada como referência própria.

5. Onde o trabalho do desenvolvedor começa

No fluxo atual, o Dev recebe da WSS um handoff documental que contém, como núcleo:

1. **Contexto de Desenvolvimento Consolidado;**
2. **Sitemap;**
3. **Briefing Final;**
4. arquivos `.md`, referências visuais e outros materiais do cliente, quando existirem.

Na intenção operacional da WSS, o Contexto de Desenvolvimento Consolidado é a principal fonte para execução.

Na prática atual do Dev, os documentos recebidos são consultados conjuntamente durante a contextualização inicial.

A partir desse handoff, a preparação técnica e a implementação passam a ser conduzidas pelo Dev.

O procedimento detalhado de leitura, organização, prompt master, `contexto.md` e `design.md` pertence ao **Manual 2 — Entrada e Contexto do Projeto**.

6. Onde o trabalho do desenvolvedor termina

A entrega técnica inicial é considerada concluída, no AS-IS, quando:

- os ajustes do cliente foram implementados;
- o site foi publicado/entregue;
- o Admin está funcional, validado e entregue.

Antes da entrega do Admin, os perfis internos já foram testados e os dados de teste são limpos.

Depois da entrega inicial, o projeto pode permanecer com o Dev para acompanhamento e manutenção pós-entrega.

Portanto, o método não termina necessariamente no primeiro publish. Ele inclui validação pós-publicação e entrega operacional do Admin.

7. Entradas principais

No nível macro, as entradas do método são:

- Contexto de Desenvolvimento Consolidado;
- Sitemap;
- Briefing Final;
- referências visuais validadas;
- identidade do cliente, quando definida;
- materiais e assets existentes;
- conteúdo e informações disponíveis;
- módulos/capacidades aplicáveis;
- decisões de kickoff;
- pendências ainda abertas;
- acessos e configurações necessários conforme o projeto.

Nem toda pendência precisa estar resolvida para toda atividade começar. A fronteira exata entre o que é obrigatório e o que pode permanecer pendente é tratada no Manual 2 e nos Manuais específicos.

8. Saídas principais

A execução pode produzir, conforme o escopo do projeto:

- site institucional implementado;
- Home e páginas internas;
- componentes e efeitos visuais;
- assets adaptados;
- Admin configurado;
- Blog configurado;
- formulários e captura de Leads;
- Chat/CTA;
- Chatbot, quando aplicável;
- perfis e acessos;
- ajustes decorrentes da revisão interna e do cliente;
- validações técnicas;
- site publicado;
- QA pós-publicação;
- acesso do cliente ao Admin;
- continuidade para manutenção ou recorrência.

Este Manual não transforma essa lista em checklist universal de conclusão. A aplicabilidade depende do projeto e do escopo.

9. Fluxo macro AS-IS

O fluxo consolidado atual pode ser representado assim:

```

Handoff documental da WSS
↓
Contextualização inicial pelo Dev
↓
ChatGPT pessoal / organização do contexto
↓
Prompt master
↓
Remix da Base Admin no Lovable
↓
Primeiros prompts e primeiros protótipos
↓
Conexão ao GitHub
↓
Clone para ambiente local
↓
Codex + Claude + Lovable
↓
Home
↓
Páginas internas
↓
Componentização / consistência / assets / conteúdo
↓
Efeitos / responsividade / performance
↓
QA interno do institucional
↓
Publicação temporária para validação do cliente
↓
Admin finalizado/configurado em paralelo
↓
Blog / Leads / Chat / Chatbot / perfis
↓
Feedback do cliente
↓
ajustes.md / rodadas de correção
↓
Auditorias e validações técnicas aplicáveis
↓
Validação final
↓
Domínio / produção
↓
QA pós-publicação
↓
Entrega do Admin
↓
Manutenção pós-entrega

```

Esse fluxo representa a jornada macro observada. Ele não deve ser interpretado como uma sequência rígida em que toda atividade só começa quando a anterior termina.

Há paralelismo real. Por exemplo, o institucional pode ser enviado para revisão do cliente enquanto o Dev continua configurando o Admin.

Para evitar ambiguidade, a coleção distingue três momentos de validação que podem se sobrepor conforme o projeto:

- 1. QA do institucional** — validação visual, responsiva e funcional do site público antes ou durante a primeira revisão externa;

2. **QA geral/operacional** — revalidação do site junto com Admin e módulos aplicáveis à medida que a camada operacional amadurece;
3. **QA pós-publicação** — validação do comportamento no domínio/ambiente final.

A numeração dos Manuais e dos SOPs é **editorial e de navegação**, não uma imposição de execução totalmente linear.

10. Papéis no fluxo

10.1 Direção / Operações WSS

Atua principalmente em:

- consolidação e encaminhamento dos insumos;
- decisões operacionais;
- esclarecimento de escopo;
- resolução de dúvidas;
- validação de possíveis mudanças de escopo;
- coordenação do projeto.

Quando o Dev identifica uma dúvida relevante, uma contradição ou uma solicitação que pode representar nova funcionalidade, a gestão pode ser acionada antes da execução.

10.2 Desenvolvedor

É o principal responsável pela execução técnica.

No AS-IS, o Dev:

- contextualiza o projeto;
- prepara o caminho inicial;
- inicia o projeto no Lovable;
- conecta GitHub e ambiente local;
- usa Codex e Claude;
- constrói e refina o site;
- configura capacidades operacionais;
- testa;
- investiga falhas;
- executa ajustes;
- participa das auditorias técnicas;
- valida produção;
- entrega o Admin.

10.3 Analise e Felipe/CEO

No fluxo atual, concentram o acesso à GoDaddy e a etapa de domínio definitivo quando essa conta é necessária.

O Dev normalmente não executa essa configuração porque não possui esse acesso.

10.4 Cliente

Participa especialmente de:

- validação de referências;
- decisões de kickoff;
- revisão do site;
- envio de feedback;
- validação factual final de conteúdos sensíveis ou específicos;
- criação do próprio acesso ao Admin.

10.5 Ferramentas de IA

Codex, Claude, Lovable e ChatGPT são ferramentas de execução e apoio.

Elas não substituem a responsabilidade humana por:

- escopo;
- fidelidade;
- decisão visual;
- risco;
- validação;
- interpretação de auditoria;
- confirmação de fatos do cliente.

11. Tipos de projeto relevantes ao método

11.1 Projeto novo no fluxo atual

O fluxo corrente parte normalmente do **remix da Base Admin no Lovable**.

A Base fornece capacidades reutilizáveis, enquanto o institucional é adaptado ao cliente.

Iniciar do zero sem a Base Admin não representa a prática corrente preferencial.

11.2 Projeto antigo / legado

Projetos antigos podem possuir arquitetura, Admin ou capacidades diferentes da geração atual.

Nesses casos, a evolução não ocorre por substituição integral automática.

O padrão observado é de **portabilidade incremental e adaptativa de capacidades**, preservando o que já funciona melhor no destino.

O procedimento pertence ao Manual 4 e ao SOP específico de portabilidade.

11.3 Projeto já publicado

Após a entrega inicial, o projeto pode continuar sob acompanhamento/manutenção.

Correções, atualizações de capacidades, conteúdo e outras evoluções passam a ocorrer sobre um projeto que já possui estado funcional a preservar.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE II — TOOLCHAIN WSS

12. Visão geral do toolchain

O método atual não depende de uma única ferramenta.

O fluxo combina ferramentas com papéis diferentes:

Ferramenta / ambiente	Papel predominante no AS-IS
ChatGPT pessoal	contextualização inicial e preparação do prompt master
Lovable	início do projeto, remix da Base Admin, primeiros protótipos e criação inicial da estrutura herdada de backend/banco/auth; no pós-criação, intervenções integradas podem ser executadas diretamente ou conduzidas pelo Codex
GitHub	ponte de versionamento e sincronização entre Lovable e ambiente local
Ambiente local	execução do projeto e base para trabalho direto com Codex/Claude
Codex	desenvolvimento direto, fidelidade visual, investigação, implementação pesada, testes e operação do Lovable em alguns casos
Claude	refinamento visual, tipografia, design, efeitos e acabamento
Supabase / Lovable Cloud	backend, banco, autenticação e recursos integrados conforme o projeto
Navegador / DevTools	validação visual/funcional e diagnóstico técnico
Awwwards	principal fonte de referências visuais
Fontes visuais auxiliares	componentes, efeitos, navegação e inspiração
SEOmator	auditoria externa SEO/GEO em casos aplicáveis
Flow	registro e contexto operacional adjacente ao desenvolvimento

A divisão não é absoluta. Uma mesma tarefa pode ser executada por mais de uma ferramenta.

13. ChatGPT pessoal

Uso atual

Antes de criar ou alterar o projeto, o Dev usa seu ChatGPT pessoal para:

- carregar documentos;
- resumir o projeto;
- organizar informações;
- compreender o caminho geral;
- avaliar por onde começar;

- preparar a instrução inicial.

Dessa etapa nasce o **prompt master**.

O Claude em terminal pode ser usado excepcionalmente nessa preparação quando o resultado obtido com ChatGPT não satisfaz.

Fronteira

Este Manual registra apenas a função do ChatGPT pessoal.

O procedimento de contextualização pertence ao Manual 2.

14. Lovable

Uso atual

O Lovable é o ponto normal de início do projeto.

É usado com frequência para:

- remix da Base Admin;
- primeiros prompts;
- primeiros protótipos;
- criação inicial da estrutura herdada de backend, banco e autenticação durante o remix da Base;
- tarefas pós-criação de backend, banco ou Supabase quando a integração da plataforma facilita, inclusive quando conduzidas pelo Codex;
- bugs funcionais;
- pequenos ajustes finais;
- operação pelo próprio Dev;
- operação indireta pelo Codex via navegador/chat em alguns casos.

Limite observado

O Dev tende a evitar o Lovable como ferramenta principal para reproduzir referências visuais complexas.

A prática antiga de desenvolver o projeto praticamente inteiro dentro do Lovable foi abandonada no fluxo atual.

Relação com outras ferramentas

O Lovable não funciona isoladamente no método atual.

Depois dos primeiros protótipos, o projeto é conectado ao GitHub e o desenvolvimento passa a ocorrer também no ambiente local.

15. GitHub

O GitHub funciona como ponte técnica entre o Lovable e o trabalho local.

Fluxo macro:

```
Lovable
↓
GitHub
↓
Projeto local
↓
Codex / Claude
↓
GitHub
↓
Lovable
```

No fluxo atual, o repositório é criado após os primeiros prompts/protótipos e clonado para a máquina do Dev.

A operação detalhada de branch, pull, commit, push, conflito e rollback pertence ao **Manual 5 — Desenvolvimento com Codex, Lovable e GitHub**.

16. Codex

Uso atual

O Codex ocupa posição central no fluxo corrente.

É usado com frequência para:

- reprodução visual com alta fidelidade;
- implementações mais pesadas;
- desenvolvimento direto no repositório;
- investigação de problemas;
- leitura e compreensão de projetos;
- análise de referências;
- análise e catalogação de imagens;
- geração contextualizada de imagens;
- testes amplos solicitados pelo Dev;
- operação do Lovable via navegador/chat em alguns casos.

Se o Dev precisasse escolher apenas uma ferramenta no fluxo atual, sua preferência declarada tenderia ao Codex.

Codex operando o Lovable

Quando o contexto já está concentrado no Codex e a tarefa faz mais sentido no Lovable, o Codex pode:

- abrir o Lovable;
- acessar o projeto;
- conversar no chat;
- acompanhar resposta e preview;
- conduzir a interação.

Essa modalidade não elimina a intervenção humana.

Pontos sensíveis como:

- credenciais;
- chaves de Supabase;
- tabelas;
- mudanças de risco;

continuam exigindo atenção e decisão humana.

17. Claude

O Claude é usado principalmente como ferramenta de refinamento.

Usos declarados:

- tipografia;
- design;
- efeitos;
- adaptação de interfaces;
- acabamento visual;
- versões mobile de composições e efeitos;
- refinamento de bases criadas pelo Codex.

O Claude não substitui o papel central do Lovable no início do projeto nem o papel de ponte do GitHub.

18. Supabase / Lovable Cloud

No fluxo atual, backend, banco, autenticação e capacidades operacionais podem envolver a infraestrutura integrada ao Lovable e ao Supabase. A criação inicial dessa estrutura herdada acontece no Lovable durante o remix da Base.

No pós-criação, tarefas de:

- backend;
- banco;

- autenticação;
- configuração relacionada;

podem ser investigadas/conduzidas pelo Codex e encaminhadas ou operadas no Lovable/Lovable Cloud quando a integração facilita ou é necessária. O uso direto do Lovable pelo Dev continua válido.

O ecossistema também aparece em diagnóstico: falhas de variável, key, conexão ou ambiente podem ser identificadas no console antes de qualquer tentativa de reescrever a funcionalidade afetada.

A arquitetura detalhada de cada módulo não pertence a este Manual.

19. Navegador e DevTools

Navegador

O navegador faz parte do trabalho manual e assistido por IA.

É usado para:

- navegar no site;
- comparar resultado e referência;
- testar interações;
- testar responsividade;
- verificar Admin;
- testar fluxos completos;
- validar produção.

DevTools / console

Não existe uma inspeção técnica rígida em DevTools para toda atividade.

O Dev recorre a console e ferramentas técnicas principalmente quando o sintoma sugere:

- configuração;
- key;
- ambiente;
- frontend;
- requests;
- erro de runtime.

O caso Mirela/Supabase é exemplo de diagnóstico em que o console ajudou a localizar a camada correta do problema.

20. Ferramentas de referência visual

20.1 Awwwards

É a principal fonte declarada de referências visuais.

A equipe normalmente seleciona entre **3 e 5 referências** modernas para o projeto, considerando UI, UX, design e desempenho percebido.

A referência principal e características de referências secundárias podem ser validadas com o cliente no kickoff.

20.2 Fontes auxiliares

Também aparecem no fluxo:

- Navbar Gallery;
- React Bits;
- 21st.dev;
- Framer;
- CodePen;
- X/Twitter;
- Instagram.

Essas fontes são usadas para encontrar:

- layouts;
- navegação;
- componentes;
- efeitos;
- animações;
- ideias visuais.

Elas não formam um catálogo obrigatório que precise ser usado em todos os projetos.

21. SEOmator

O SEOmator foi confirmado como ferramenta externa utilizada em auditoria SEO/GEO.

No fluxo observado, seu resultado é tratado como **entrada para investigação**, não como ordem automática de alteração.

Uma auditoria externa pode:

- apontar problema real;
- refletir limitação do crawler;
- estar baseada em versão anterior;
- exigir interpretação antes de qualquer edição.

O procedimento de triagem e tratamento da auditoria pertence ao Manual 6 e ao SOP correspondente.

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Não há evidência de uso obrigatório do SEOmator em toda entrega nem de score mínimo universal que funcione como gate WSS.

Heurística pessoal declarada pelo Dev — não normativa

Na prática pessoal do Dev, a faixa verde é buscada como referência; um score em torno de 90 é considerado bom e 95+ é uma meta desejada quando viável. Essa heurística não é regra WSS, não é requisito de entrega e não substitui a análise técnica dos achados.

22. Flow e ferramentas operacionais adjacentes

O Flow não é o ambiente principal de desenvolvimento do código.

Ele aparece como camada operacional adjacente, por exemplo em:

- registro de projeto;
- tarefas e acompanhamento interno;
- registros operacionais;
- armazenamento de credenciais internas conforme o fluxo atual.

A documentação completa do Processo Sites WSS/Flow permanece fora do escopo deste Manual.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE III — CRITÉRIOS DE ESCOLHA DAS FERRAMENTAS

23. Regra prática atual

Não existe hoje uma matriz formal e obrigatória que determine ferramenta por tipo de tarefa.

A escolha é pragmática e depende de:

- natureza da tarefa;
- qualidade do resultado anterior;
- integração disponível;
- nível de fidelidade visual desejado;
- necessidade de trabalhar diretamente no código;
- necessidade de investigar;
- camada afetada;
- contexto já carregado em determinada IA;
- risco da alteração.

AUSÊNCIA ATUAL DE PADRONIZAÇÃO

Não existe uma regra formal do tipo:

“toda tarefa da categoria X deve obrigatoriamente ser executada pela ferramenta Y”.

O método atual depende de julgamento do Dev.

24. Critérios observados de roteamento

Preferência por Lovable

O Lovable tende a ser escolhido quando:

- o projeto está começando e a Base precisa ser remixada;
- a criação inicial da estrutura herdada de backend/banco/auth acontece no projeto remixado;
- no pós-criação, a integração com backend/banco/Supabase torna a execução no Lovable/Lovable Cloud mais adequada;
- o problema é funcional;
- há pequenos ajustes finais;
- a tarefa já faz mais sentido dentro do contexto do Lovable.

Preferência por Codex

O Codex tende a ser escolhido quando:

- é necessária maior fidelidade visual;
- a implementação é mais pesada;
- é necessário trabalhar diretamente no código;
- há necessidade de investigação;
- no pós-criação, é útil investigar ou conduzir uma intervenção de backend/banco/auth e, quando necessário, operar/orquestrar o Lovable;
- é preciso analisar uma referência;
- a tarefa exige testes ou inspeções mais amplas;
- o contexto já está concentrado no Codex.

Preferência por Claude

O Claude tende a ser escolhido quando:

- o problema é de refinamento visual;
- tipografia, composição ou acabamento precisam melhorar;
- um efeito precisa de adaptação;
- uma solução desktop precisa de leitura específica para mobile.

Preferência por abordagem manual/humana

A intervenção humana permanece central quando é necessário:

- julgar se ficou bom;
- comparar referência e resultado;
- avaliar fidelidade;
- decidir se uma adaptação faz sentido;
- proteger escopo;
- decidir diante de contradição;
- validar risco;
- impedir que IA altere áreas já aprovadas;
- confirmar informação sensível ou factual.

25. Trabalho manual do Dev como camada do método

O método atual não é um fluxo “IA executa e humano aceita”.

A camada manual é composta principalmente por:

- julgamento;
- navegação;
- comparação;
- cliques;

- testes;
- decisão de fidelidade;
- validação de comportamento;
- correção do entendimento da IA;
- controle de escopo;
- decisão de risco.

Essa camada é especialmente importante porque várias partes do método atual não possuem critérios formais completamente automatizáveis.

Exemplos:

- qualidade visual;
 - percepção de performance;
 - decisão de preservar ou descartar um efeito;
 - escolha de qual ferramenta usar;
 - definição de quando um resultado já está bom o suficiente para avançar.
-

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE IV — GOVERNANÇA E LIMITES

26. Fontes de conhecimento do método

A documentação atual foi consolidada a partir de diferentes tipos de fonte:

- Direção/Operações;
- Dev;
- demonstração prática;
- projetos reais;
- código;
- GitHub;
- Lovable;
- artefatos técnicos;
- validações dirigidas.

Uma peculiaridade encontrada em um projeto não deve ser convertida automaticamente em regra geral.

Quando necessário, os achados podem ser classificados como:

- **CONFIRMADO;**
- **OBSERVADO;**
- **INFERIDO;**
- **DEPRECIADO;**
- **PENDENTE.**

Essa classificação serve para governar a documentação, não para transformar o Manual em relatório arqueológico.

27. Decisões humanas que permanecem centrais

Mesmo com forte uso de IA, permanecem humanas decisões como:

- escopo;
- prioridade;
- mudança de escopo;
- direção visual;
- fidelidade;
- adequação da referência;
- risco de alteração;
- validação factual;
- interpretação de auditorias;
- aprovação para publicação;
- contingência diante de falha.

Nenhuma ferramenta deve ser tratada como autoridade absoluta apenas por produzir uma análise, um diff ou uma sugestão.

28. NORMA TÉCNICA VIGENTE

A autoridade canônica vigente para SEO, AEO e GEO 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` está superado como autoridade autônoma e permanece apenas como baseline histórico/proveniência quando necessário.

As normas devem ser respeitadas quando aplicáveis independentemente de a implementação ocorrer por Lovable, Codex, Claude Code ou outra ferramenta.

Entretanto:

a norma técnica não deve ser usada para reescrever retroativamente o AS-IS documentado.

Nos Manuais seguintes, sempre que uma norma complementar a prática atual, as duas camadas devem permanecer explicitamente diferenciadas.

29. AUSÊNCIAS ATUAIS DE PADRONIZAÇÃO relevantes para esta visão geral

No nível macro, o AS-IS não possui:

- matriz formal obrigatória de escolha de ferramenta;
- automação que determine sozinha Lovable × Codex × Claude;
- ritual universal de trabalho idêntico para todos os tipos de projeto;
- uso obrigatório de todas as ferramentas auxiliares;
- score SEO/performance externo único como gate geral;
- eliminação completa do julgamento humano.

Essas ausências não impedem a reprodução do método, desde que o executor compreenda quais decisões exigem julgamento.

30. Práticas atuais que explicam o desenho do toolchain

Duas mudanças históricas ajudam a explicar o fluxo atual:

1. **desenvolver praticamente tudo apenas dentro do Lovable** deixou de ser a abordagem predominante;

- 2. iniciar todo projeto do zero** foi substituído, no fluxo corrente, pelo uso da Base Admin como ponto de partida.

Por isso o toolchain atual é híbrido:

```
Lovable  
+ GitHub  
+ ambiente local  
+ Codex  
+ Claude  
+ validação humana
```

O detalhamento das práticas depreciadas pertence ao Manual 8.

WSS · MÉTODO DE DESENVOLVIMENTO DE SITES

PARTE V — EXEMPLO REAL E NAVEGAÇÃO DA COLEÇÃO

31. Exemplo real resumido — Dr. Guilherme Mendonça

O projeto do Dr. Guilherme Mendonça é usado como principal caso representativo do fluxo atual.

Ele exemplifica:

- início por remix da Base Admin;
- primeiros passos no Lovable;
- integração com GitHub;
- trabalho local;
- uso combinado de Codex e Claude;
- construção visual;
- configuração do Admin;
- QA;
- auditorias de SEO/AEO/GEO;
- publicação e entrega.

Exclusão explícita

Em determinado momento houve um esboço antecipado antes de todas as definições do cliente estarem concluídas.

Essa ocorrência foi tratada como exceção operacional e **não integra o Método WSS**.

O estudo completo do caso pertence ao Manual 8.

32. Relação com os demais Manuais

Manual 2 — Entrada e Contexto do Projeto

Explica como transformar o handoff em contexto utilizável, prompt master e referências persistentes.

Manual 3 — Construção Visual

Explica referências, Home, páginas internas, componentização, Design System, conteúdo, assets, efeitos, responsividade, performance e acessibilidade AS-IS.

Manual 4 — Projeto-Base, Admin e Funcionalidades

Explica Base Admin, módulos, Blog, Leads, Chat/CTA, Chatbot, autenticação, perfis e portabilidade para projetos existentes.

Manual 5 — Desenvolvimento com Codex, Lovable e GitHub

Explica operacionalmente repositório, trabalho local, sincronização, modelos de uso do Codex/Lovable, conflitos e rollback.

Manual 6 — Testes, QA, Debugging, Segurança e SEO

Explica validação, regressão, troubleshooting, auditorias, SEO/AEO/GEO e segurança.

Manual 7 — Publicação, Entrega e Pós-Publicação

Explica domínio temporário, produção, QA pós-publicação, entrega do Admin e continuidade.

Manual 8 — Boas Práticas, Anti-Patterns, Práticas Depreciadas e Casos Reais

Concentra conhecimento acumulado, problemas conhecidos e estudos de caso.

33. SOPs relacionados

Este Manual não gera SOP próprio.

Ele funciona como pré-leitura e mapa para os SOPs da coleção, principalmente:

- SOP-001 — Contextualizar um Novo Projeto WSS;
 - SOP-002 — Iniciar Tecnicamente um Projeto WSS: Base → Lovable → GitHub → Ambiente Local;
 - SOP-003 — Construir e Refinar a Home a partir das Referências;
 - SOP-005 — Configurar Admin e Módulos Aplicáveis;
 - SOP-008 — Executar QA Visual, Funcional e Responsivo;
 - SOP-012 — Preparar, Publicar, Validar Produção e Entregar Admin.
-

34. Critério de conclusão deste Manual

Este Manual cumpre seu objetivo quando o leitor consegue responder:

1. o que é o Método de Desenvolvimento de Sites WSS;
 2. como ele se distingue do Processo Sites WSS;
 3. onde o trabalho do Dev começa e termina;
 4. qual é o fluxo macro atual;
 5. quais são as principais ferramentas;
 6. para que cada ferramenta tende a ser usada;
 7. como elas se conectam;
 8. quais decisões permanecem humanas;
 9. quais ausências de padronização não devem ser preenchidas com um processo ideal;
 10. em qual Manual/SOP procurar o procedimento detalhado.
-

35. 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_Development_Sites_Plano_Conducao_v0.3.md ;
5. WSS_Metodo_Development_Sites_Indice_Mestre_v0.1_DRAFT.md .

Contexto adicional utilizado apenas na fronteira do Processo:

- Roadmap_Sites_WSS_Modelo_Padrao_v0.2_FINAL.md .

Normas técnicas são referências próprias e devem permanecer separadas do relato AS-IS.

Autoridade normativa SEO/AEO/GEO vigente:

- WSS_Padrao_Tecnico_Operacional_SEO_AEO_GEO_v0.2_FINAL.md .

Status: FINAL — APROVADO