

Sites e aplicativos com IA

Como usar IA para projetar, prototipar e desenvolver produtos digitais sem a aparência genérica e com qualidade, acessibilidade e segurança.



D.B. Conhecimento

Coleção Inteligência Aplicada · Volume 11

■ SITES E APLICATIVOS COM IA: DO DESIGN AO CÓDIGO CONFIÁVEL

Como usar IA para projetar, prototipar e desenvolver produtos digitais sem a aparência genérica e com qualidade, acessibilidade e segurança.

1ª edição · outubro de 2026 · Autoria: D.B. Soluções Digitais.

■ USO E DISTRIBUIÇÃO

Material educativo de distribuição gratuita. É permitido ler, imprimir e compartilhar este arquivo sem alterações, inclusive dentro de empresas. A venda é proibida. Trechos podem ser citados com indicação da fonte: "D.B. Soluções Digitais, D.B. Conhecimento".

■ AVISO

O conteúdo tem finalidade educativa e não substitui consultoria jurídica, contábil, tributária ou de segurança da informação. Estudos de caso e exemplos numéricos são ilustrativos, com empresas e valores fictícios. Marcas e produtos citados pertencem aos respectivos titulares; a menção não indica parceria nem recomendação comercial. Ferramentas de IA mudam rapidamente: confirme recursos, preços e termos nas fontes oficiais.

■ CRÉDITOS

Texto, diagramas e projeto gráfico: D.B. Soluções Digitais. Imagens geradas para este projeto com inteligência artificial. Tipografia: Satoshi (Indian Type Foundry, licença ITF FFL), Source Serif 4, Manrope, Instrument Serif e JetBrains Mono (SIL Open Font License).

© 2026 D.B. Soluções Digitais. Além do código.

Rápido não é o mesmo que pronto

Nunca foi tão rápido criar um site ou um aplicativo. Com um assistente de programação, uma descrição em linguagem natural vira uma página funcional em minutos; uma captura de tela vira um protótipo; uma ideia de aplicativo ganha telas, banco de dados e tela de pagamento em poucos dias. Bibliotecas abertas sobre IA mostram pessoas sem formação técnica lançando produtos digitais com apoio dessas ferramentas.

A velocidade, porém, esconde uma armadilha. Um site gerado em minutos costuma se parecer com milhares de outros gerados da mesma forma. Um protótipo que funciona na demonstração pode não funcionar com usuários reais, em celulares antigos, com conexões lentas ou com leitores de tela. Um aplicativo montado às pressas pode expor dados de clientes, usar bibliotecas com vulnerabilidades ou depender de componentes cuja licença não permite o uso pretendido.

Esta apostila trata do que separa um produto digital feito com IA de um produto digital bem feito com IA: direção de design, processo, arquitetura, testes, acessibilidade, desempenho, segurança e cuidados com licenças e dados pessoais. Ela serve a quem desenvolve e também a quem contrata: gestores que precisam avaliar propostas, prazos e entregas em um mercado em que “fazer um site” pode significar coisas muito diferentes.

O que você vai encontrar

—	Apresentação	3
01	O problema: rápido não é o mesmo que pronto	7
	Por que sites feitos com IA parecem iguais	7
	Os sinais de uma interface feita sem direção	8
	Protótipo não é produto	8
02	Do pedido ao produto: o processo	10
	As etapas	10
	Onde a IA ajuda em cada etapa	12
	A descoberta não pode ser pulada	12
	Como comparar propostas	13
03	Direção de design: referências e repertório	14
	Montando o repertório	14
	Tipografia e hierarquia	16
	Referência não é cópia	17
	Habilidades de “bom gosto”	17
04	Protótipos a partir de imagens e descrições	18
	Da imagem ao código	18
	O ciclo de verificação visual	18
	Protótipos para validar, não para lançar	19
05	Desenvolvimento assistido por IA	21
	Preparar o projeto para o assistente	21
	Tarefas pequenas e verificáveis	21
	Escolha da tecnologia	22
	Aplicativos móveis: o que muda	22
	Revisão de código	23
06	Qualidade: testes, acessibilidade e desempenho	25
	Testes em camadas	25
	Desempenho	27
	Fundamentos de busca	28
	Responsividade e verificação real	28

07	Segurança, licenças e dados pessoais	29
	Riscos de segurança mais frequentes	29
	Licenças de software	30
	Dados pessoais em sites e aplicativos	30
	Hospedagem, cópias e monitoramento	31
	Checklist de lançamento	32
08	Implantação de um projeto e indicadores	33
	Um projeto típico em fases	33
	Indicadores	34
09	Estudo de caso ilustrativo e exercícios	35
10	Como a D.B. pode atuar e conclusão	37
	Conclusão	37
—	Glossário	38
—	Referências	39

Ao final desta apostila, você será capaz de:

- 01 Reconhecer os sinais de interfaces genéricas e como evitá-los.
- 02 Montar um repertório de referências e transformá-lo em direção de design.
- 03 Prototipar a partir de imagens e descrições com verificação visual.
- 04 Organizar projeto, arquitetura e instruções para desenvolvimento assistido por IA.
- 05 Aplicar testes, revisão, acessibilidade (WCAG) e métricas de desempenho.
- 06 Tratar segurança, licenças e propriedade intelectual no código gerado.

Para quem é este material. empreendedores com ideias de produtos digitais; designers e desenvolvedores; gestores que contratam sites e sistemas.

Como aproveitar melhor. Leia um capítulo por vez, faça os exercícios com um processo real da sua empresa e use os checklists como roteiro de reunião. Os estudos de caso são ilustrativos: servem para mostrar o raciocínio, não para prometer resultados.

O problema: rápido não é o mesmo que pronto

A IA tornou barato produzir a primeira versão. A qualidade continua exigindo método.

■ Por que sites feitos com IA parecem iguais

Peça a qualquer modelo de linguagem “uma página moderna para minha empresa” e o resultado tende à média de tudo o que ele viu: um título grande centralizado, um gradiente roxo ou azul, botões com brilho, três cartões com ícones genéricos lado a lado, depoimentos inventados e um rodapé padrão. Não é um defeito do modelo; é o resultado esperado de um pedido sem direção. O modelo escolhe o caminho mais provável, e o caminho mais provável é o comum.

Para uma empresa, parecer genérica tem custo: a marca não se diferencia, a mensagem não se fixa e o visitante sai sem lembrar de quem era o site.

■ Os sinais de uma interface feita sem direção

Tabela 1.1 – Sinais comuns de interfaces genéricas geradas sem direção de design

Sinal	Por que acontece	Alternativa
Gradientes e brilhos em tudo	Estética padrão de muitos modelos e modelos prontos	Paleta da marca, com acentos localizados
Três cartões de benefícios com ícones	Estrutura mais comum em páginas de produto	Organizar pelo raciocínio do cliente, não pelo padrão
Textos vagos (“soluções inovadoras”)	Falta de briefing e de conteúdo real	Texto escrito a partir das perguntas do cliente
Depoimentos e números inventados	O modelo preenche espaços vazios	Só conteúdo real, autorizado e comprovável
Tipografia padrão sem hierarquia	Fontes do sistema e tamanhos uniformes	Pares tipográficos escolhidos e escala definida
Tudo centralizado e com o mesmo peso	Ausência de composição	Hierarquia clara: o que o visitante deve ver primeiro
Imagens genéricas de banco	Falta de direção de arte	Imagens próprias ou geradas com guia de estilo

■ Protótipo não é produto

Um protótipo serve para validar uma ideia: o fluxo faz sentido? o cliente entende a proposta? Um produto precisa funcionar todos os dias, para pessoas diferentes, em condições adversas, com dados reais e por anos. Entre um e outro estão etapas que a IA acelera, mas não elimina:

- tratamento de erros e de casos extremos;
- acessibilidade para pessoas com deficiência;
- desempenho em celulares comuns e conexões lentas;
- segurança de dados, autenticação e permissões;
- testes automatizados e manutenção;
- conformidade com a LGPD em formulários e cadastros;
- documentação para que outra pessoa consiga manter o sistema.

! ATENÇÃO**O custo escondido da pressa**

Sistemas gerados rapidamente, sem testes nem revisão, costumam acumular o que os desenvolvedores chamam de dívida técnica: soluções provisórias que tornam cada mudança futura mais lenta e arriscada. O que pareceu barato no lançamento pode se tornar caro na primeira manutenção.

Do pedido ao produto: o processo

Cada etapa existe para responder a uma pergunta antes que ela se torne um problema.

■ As etapas

O processo de criação de um produto digital tem etapas que existem há muito tempo, antes da IA. O que mudou é a velocidade de algumas delas: protótipos e código ficaram muito mais rápidos. As etapas de decisão e de verificação, porém, continuam exigindo tempo e atenção humana. Pular a descoberta ou os testes porque o código ficou pronto em minutos é a forma mais comum de transformar velocidade em retrabalho. A sequência abaixo mostra o caminho completo.



Figura 2.1 – Processo de criação de um site ou aplicativo com apoio de IA

■ Onde a IA ajuda em cada etapa

Tabela 2.1 – Contribuições da IA ao longo do processo

Etapa	Contribuição da IA	Decisão humana
Descoberta	Organizar entrevistas, levantar perguntas frequentes, analisar concorrentes	Definir objetivos e prioridades
Arquitetura	Sugerir estruturas e fluxos a partir do conteúdo	Validar com o público
Direção de design	Extrair padrões de referências, propor variações	Escolher a direção
Protótipo	Gerar telas a partir de descrições e imagens	Testar e decidir
Desenvolvimento	Escrever código, testes e documentação	Revisar e aprovar
Qualidade	Executar testes, apontar problemas de acessibilidade e desempenho	Priorizar correções
Evolução	Analisar dados de uso, sugerir melhorias	Decidir o roteiro

■ A descoberta não pode ser pulada

A tentação de ir direto ao código é grande quando a IA entrega telas em minutos. Mas um site sem descoberta é bonito e inútil: não sabe a quem fala, nem o que o visitante deve fazer depois de chegar. As perguntas essenciais da descoberta são simples:

- quem visita o site e o que procura?
- qual ação queremos que o visitante realize?
- que conteúdo real temos: textos, imagens, casos, provas?
- que sistemas o site precisa conectar: formulários, CRM, agenda, pagamentos?
- como saberemos que o site está cumprindo seu papel?

■ Como comparar propostas

Para quem contrata, a popularização da IA trouxe propostas com preços e prazos muito diferentes para “o mesmo site”. A diferença costuma estar no que está incluído. Perguntas objetivas ajudam a comparar:

Tabela 2.2 – Perguntas para comparar propostas de sites e aplicativos

Tema	Pergunta ao fornecedor
Descoberta	Haverá entrevistas e definição de objetivos antes do design?
Conteúdo	Quem escreve os textos? Haverá revisão? Há uso de conteúdo inventado?
Design	Haverá sistema visual próprio ou modelo pronto adaptado?
Tecnologia	Qual tecnologia será usada e por quê? Quem consegue manter depois?
Integrações	Formulários, agenda, CRM e pagamentos estão incluídos?
Qualidade	Quais testes serão feitos? Há auditoria de acessibilidade e desempenho?
Segurança e LGPD	Como os dados dos formulários são protegidos? Há política de privacidade?
Propriedade	Domínio, hospedagem, código e contas ficam em nome de quem?
Manutenção	O que acontece depois do lançamento? Qual é o custo de ajustes?
Documentação	Haverá documentação e treinamento para editar conteúdo?

Propostas mais baratas não são necessariamente piores, e as mais caras não são necessariamente melhores. O importante é saber exatamente o que está sendo comprado.

Direção de design: referências e repertório

A IA não sabe o que é bonito para a sua marca. Ela precisa ver exemplos e regras.

■ Montando o repertório

Designers experientes têm repertório: centenas de referências vistas, analisadas e guardadas. Para que a IA produza algo com personalidade, é preciso fornecer esse repertório. Existem galerias e índices curados de bons sites, bibliotecas de componentes, coleções de fontes e ferramentas de design. Reunir de cinco a dez referências bem escolhidas, com uma explicação do que se gosta em cada uma, muda completamente o resultado.



Figura 3.1 — Direção de design começa no papel: referências, hierarquia e escolhas antes do código

Da referência ao sistema visual

Gostar de um site não basta; é preciso entender **por que** ele funciona. Para cada referência, registre:

- **tipografia:** famílias, pesos, tamanhos e espaçamentos;
- **cores:** base, texto, acentos e como são usadas;
- **composição:** grade, margens, alinhamentos e uso do espaço vazio;
- **componentes:** botões, cartões, menus, formulários;
- **movimento:** transições, animações e interações;
- **tom:** o que a página comunica sem palavras.

A partir dessa análise, define-se o **sistema visual** do projeto: um conjunto de decisões registradas como tokens de design, isto é, variáveis nomeadas para cores, tamanhos, espaçamentos, raios e sombras. Ferramentas de IA conseguem extrair uma primeira proposta de tokens a partir de capturas de tela de referências, que depois é ajustada à identidade da marca.

Tabela 3.1 — Exemplo de tokens de design

Token	Valor	Uso
cor-fundo	#0E0E0E	Fundo principal no tema escuro
cor-texto	#F5F3EF	Texto principal
cor-acento	#FF9233	Destaques, links e foco
fonte-titulo	Satoshi, 700	Títulos
fonte-texto	Manrope, 400	Texto corrido
raio-cartao	16 px	Cartões e painéis
espaco-secao	96 px	Distância entre seções no desktop

■ Tipografia e hierarquia

Boa parte da percepção de qualidade de uma interface vem da tipografia. Ellen Lupton, em *Pensar com tipos*, mostra como escolhas de família, tamanho, espaçamento e alinhamento organizam a leitura. Algumas orientações práticas para sites:

- **poucas famílias:** em geral, uma para títulos e outra para textos, ou uma só com pesos variados;
- **escala definida:** tamanhos que crescem em proporção constante, aplicados de forma consistente;
- **texto corrido confortável:** tamanho em torno de 16 a 18 pixels, linhas com cerca de 60 a 80 caracteres e entrelinha generosa;
- **contraste de peso:** títulos visivelmente mais fortes que o texto, sem precisar de cores chamativas;
- **hierarquia por espaço:** distância maior antes de um título do que depois dele, para que o título se ligue ao conteúdo que apresenta.

Quando a hierarquia está bem resolvida, o visitante entende a página antes mesmo de ler: o que é mais importante, o que é detalhe e onde agir.

■ Referência não é cópia

Estudar sites de outras empresas é parte normal do trabalho de design. Copiar textos, imagens, logotipos, ilustrações ou código de outras empresas não é. O caminho correto é extrair princípios, como hierarquia, ritmo, uso de espaço e estrutura de navegação, e aplicá-los com conteúdo, identidade e implementação próprios. Isso vale também para ferramentas de engenharia reversa, capazes de analisar como uma funcionalidade foi construída: elas ajudam a aprender e a construir a sua versão, não a reproduzir o produto de outra empresa.

■ Habilidades de “bom gosto”

Uma prática que tem se espalhado é registrar em arquivos os princípios de design e os sinais de interface genérica, para que o assistente de programação os leia antes de cada trabalho. Esses arquivos, organizados como habilidades, incluem as referências escolhidas, as regras do sistema visual e uma lista do que evitar. O resultado é um assistente que começa cada tela já dentro da direção do projeto.



CHECKLIST

Briefing de design para o assistente

- ☐ Objetivo da página e ação principal esperada do visitante.
- ☐ Público e o que ele precisa entender primeiro.
- ☐ Cinco a dez referências, com o que se deve aproveitar de cada uma.
- ☐ Tokens do sistema visual: cores, tipografia, espaços, raios.
- ☐ Lista de padrões a evitar.
- ☐ Conteúdo real disponível; nada de textos ou números inventados.
- ☐ Requisitos de acessibilidade e de desempenho.

Protótipos a partir de imagens e descrições

Mostrar é mais rápido do que explicar. A IA transforma referências em telas navegáveis.

■ Da imagem ao código

Ferramentas de código aberto, como o projeto screenshot-to-code, e os próprios assistentes de programação conseguem transformar uma captura de tela, um desenho feito à mão ou um arquivo de design em código funcional. Os melhores resultados acontecem quando:

- a imagem é clara e mostra uma tela inteira, com hierarquia visível;
- o pedido indica a tecnologia desejada e o sistema visual do projeto;
- o assistente pode abrir o resultado em um navegador, comparar com a referência e corrigir as diferenças;
- logotipos e imagens são os oficiais do projeto, e não recriações aproximadas.

■ O ciclo de verificação visual

Um assistente de programação que gera uma tela sem vê-la renderizada trabalha às cegas. O ciclo mais produtivo inclui a verificação visual:

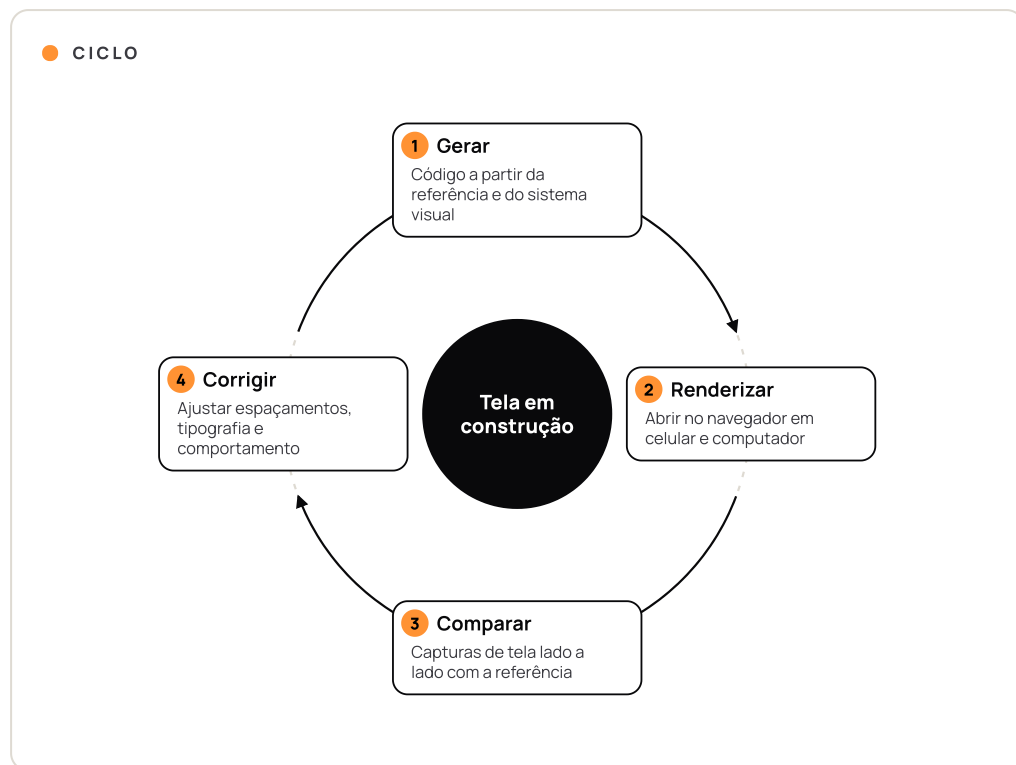


Figura 4.1 – Ciclo de prototipagem com verificação visual

Ferramentas de automação de navegador permitem que o assistente capture telas em diferentes tamanhos e compare com a referência. Ainda assim, a avaliação final é humana: alguém precisa olhar as telas reais, em aparelhos reais, e decidir se estão boas.

■ Protótipos para validar, não para lançar

O valor do protótipo está em testar a ideia com pessoas antes de investir no desenvolvimento completo. Cinco conversas com usuários representativos, observando-os tentar realizar tarefas no protótipo, revelam a maior parte dos problemas de compreensão, como mostram décadas de prática em usabilidade, sistematizadas por autores como Jakob Nielsen e Steve Krug. A IA reduz o custo do protótipo; a observação de pessoas reais continua insubstituível.

**DICA PRÁTICA****Protótipo descartável**

Combine com a equipe e com o cliente que o protótipo é descartável. Isso permite usar a IA com liberdade para explorar alternativas, sem a pressão de que aquele código precise virar o produto. Depois da validação, o desenvolvimento começa com a estrutura adequada.

Desenvolvimento assistido por IA

O assistente escreve código rápido. A equipe garante que ele seja o código certo.

■ Preparar o projeto para o assistente

Assistentes de programação trabalham melhor em projetos organizados. Antes de pedir funcionalidades, prepare:

- **arquivo de instruções do projeto**, como o AGENTS.md, com a arquitetura, os comandos de execução e de teste, as convenções de código e as regras de segurança;
- **mapa de arquitetura** atualizado, que mostre os componentes e como se conectam. Há habilidades que geram esse mapa automaticamente a partir do código;
- **testes automatizados** que o assistente possa executar para verificar o próprio trabalho;
- **controle de versões**, com cada mudança registrada e revisável.

■ Tarefas pequenas e verificáveis

Pedidos grandes, como “faça o sistema de agendamento completo”, geram muito código difícil de revisar. Pedidos pequenos, com critério de aceite claro, geram mudanças que podem ser revisadas, testadas e revertidas.

Tabela 5.1 — Pedidos amplos e pedidos verificáveis

Pedido amplo	Pedidos verificáveis
Faça o sistema de agendamento	1. Criar a tabela de horários com os campos X, Y e Z e testes. 2. Criar a tela de escolha de horário com os estados vazio, carregando e erro. 3. Integrar a confirmação por e-mail com envio em fila.
Melhore o desempenho do site	Reduzir o tamanho das imagens da página inicial e medir o LCP antes e depois
Deixe o site acessível	Corrigir os problemas apontados pela auditoria automática e testar a navegação apenas por teclado

■ Escolha da tecnologia

A tecnologia deve atender ao projeto, à equipe que vai mantê-lo e ao orçamento. Para sites de conteúdo, geradores de sites estáticos e frameworks com renderização no servidor oferecem bom desempenho e baixo custo de hospedagem. Para aplicativos móveis, frameworks multiplataforma permitem uma única base de código para Android e iOS. Serviços de backend gerenciados aceleram cadastro de usuários, banco de dados e armazenamento. Bibliotecas de pagamento e de assinatura simplificam a cobrança.

Pilhas tecnológicas famosas por terem sido usadas em aplicativos de sucesso são boas referências, mas não garantem resultado. O sucesso desses produtos dependeu de mercado, produto, distribuição e execução, e não apenas das ferramentas.

■ Aplicativos móveis: o que muda

Aplicativos para celular compartilham a maior parte dos princípios de sites, mas trazem particularidades que a IA não resolve sozinha:

- **lojas de aplicativos:** Google Play e App Store têm diretrizes de revisão, exigências de privacidade e prazos de aprovação; rejeições são comuns em primeiras versões;
- **declarações de privacidade:** as lojas exigem informar quais dados o aplicativo coleta e para quê, o que precisa ser coerente com a política de privacidade e com a LGPD;
- **atualizações:** correções dependem de publicar novas versões e de os usuários atualizarem, o que torna os testes antes do lançamento ainda mais importantes;
- **condições do aparelho:** conexão instável, pouca memória, bateria e diferentes tamanhos de tela exigem testes em aparelhos reais;
- **cobrança:** assinaturas e compras dentro do aplicativo seguem regras específicas das lojas, inclusive sobre taxas e formas de pagamento.

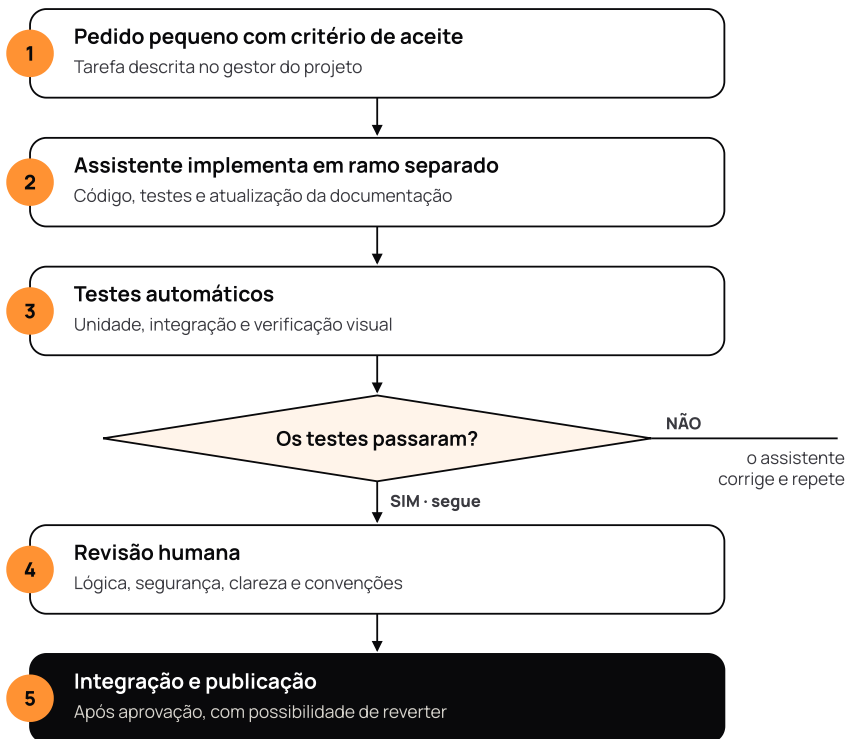
Bibliotecas abertas sobre IA têm mostrado pessoas que construíram aplicativos de sucesso com equipes pequenas. Esses casos são inspiradores, mas não devem ser lidos como fórmula: por trás de cada produto que deu certo houve pesquisa de mercado, testes, distribuição e muitas versões que não deram certo.

■ Revisão de código

Todo código gerado por IA deve ser revisado como se tivesse sido escrito por um colega novo e muito produtivo: com atenção à lógica, à segurança, à clareza e à aderência às convenções do projeto. Pontos de atenção específicos:

- **dependências inventadas:** modelos podem sugerir bibliotecas que não existem ou com nomes parecidos com bibliotecas reais; atacantes já exploram esse comportamento publicando pacotes maliciosos com esses nomes. Confirme a existência, a reputação e a manutenção de cada pacote antes de instalar;
- **soluções excessivamente complexas** para problemas simples;
- **tratamento de erros ausente** ou genérico;
- **segredos no código**, como chaves de API;
- **testes que testam pouco**, escritos apenas para passar.

● FLUXOGRAMA

**Figura 5.1** – Fluxo de uma mudança de código com assistente de IA

Qualidade: testes, acessibilidade e desempenho

Qualidade não é uma etapa no fim. É um critério em cada entrega.

■ Testes em camadas

Mike Cohn popularizou a ideia da pirâmide de testes: muitos testes rápidos de unidade na base, menos testes de integração no meio e poucos testes de ponta a ponta no topo, que simulam o usuário no navegador. Assistentes de IA escrevem testes com rapidez, o que torna essa estrutura mais acessível a equipes pequenas.



Figura 6.2 – Camadas de testes de um produto digital

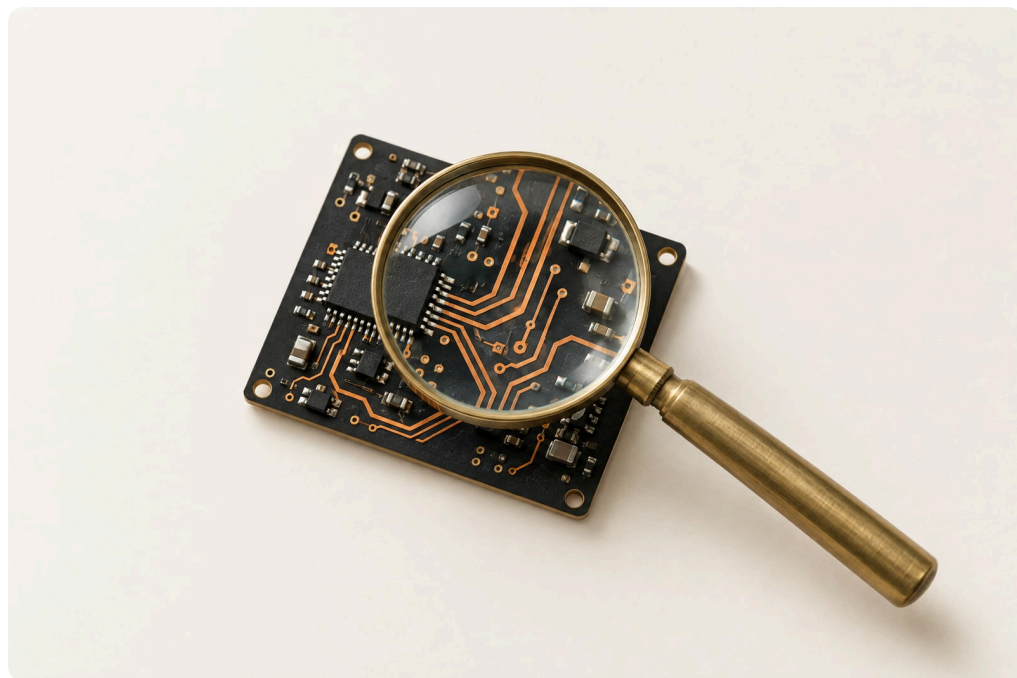


Figura 6.1 — Qualidade se verifica de perto: código, interface e comportamento real

Acessibilidade

As Diretrizes de Acessibilidade para Conteúdo Web, as WCAG, publicadas pelo W3C, são a principal referência internacional. A versão 2.2, de 2023, organiza os critérios em quatro princípios: perceptível, operável, compreensível e robusto. No Brasil, a Lei Brasileira de Inclusão também trata da acessibilidade em sites. Itens essenciais:

**CHECKLIST****Acessibilidade essencial**

- ☐ Contraste suficiente entre texto e fundo.
- ☐ Navegação completa pelo teclado, com foco visível.
- ☐ Textos alternativos em imagens informativas.
- ☐ Títulos e estrutura de cabeçalhos em ordem lógica.
- ☐ Formulários com rótulos, instruções e mensagens de erro claras.
- ☐ Legendas em vídeos com fala.
- ☐ Respeito à preferência do sistema por redução de movimento.
- ☐ Página utilizável com zoom de 200%.

Ferramentas automáticas encontram parte dos problemas; outra parte só aparece em testes manuais, como navegar por teclado e usar um leitor de tela.

■ Desempenho

O Google definiu as Core Web Vitals, métricas de experiência do usuário relacionadas ao carregamento, à responsividade e à estabilidade visual. Os limites considerados bons, segundo a documentação oficial, são:

Tabela 6.1 — Core Web Vitals e limites considerados bons

Métrica	O que mede	Limite bom
LCP (Largest Contentful Paint)	Tempo até o maior elemento visível carregar	Até 2,5 segundos
INP (Interaction to Next Paint)	Tempo de resposta às interações	Até 200 milissegundos
CLS (Cumulative Layout Shift)	Estabilidade visual durante o carregamento	Até 0,1

Os vilões mais comuns do desempenho são imagens grandes, fontes em excesso, scripts de terceiros e animações pesadas. Medir antes e depois de cada mudança evita surpresas.

■ Fundamentos de busca

Um site bem feito também precisa ser encontrado. Os mecanismos de busca valorizam páginas úteis, rápidas, acessíveis e bem estruturadas, o que coincide em grande parte com as boas práticas deste capítulo. Itens técnicos essenciais:

Tabela 6.2 – Fundamentos técnicos de otimização para mecanismos de busca

Item	Boa prática
Título da página	Único, descritivo e com o tema principal da página
Descrição	Resumo claro do conteúdo, que convide ao clique
Estrutura de títulos	Um título principal por página e subtítulos em ordem
Endereços	Curtos, legíveis e estáveis
Imagens	Texto alternativo, tamanho adequado e formatos modernos
Mapa do site	Arquivo que lista as páginas para os buscadores
Dados estruturados	Marcação que descreve organizações, artigos, produtos e eventos
Desempenho	Core Web Vitals dentro dos limites considerados bons
Conteúdo	Textos que respondem às perguntas reais do público

A IA ajuda a sugerir títulos e descrições, a revisar a estrutura e a encontrar páginas sem texto alternativo. A escrita do conteúdo, porém, deve partir do que a empresa realmente sabe e oferece; páginas geradas em massa sem valor costumam ser ignoradas pelos buscadores e pelos visitantes.

■ Responsividade e verificação real

Sites precisam funcionar em telas de todos os tamanhos. A verificação deve incluir, no mínimo, uma tela de celular comum, um tablet e um computador, e, sempre que possível, aparelhos reais, porque emuladores não reproduzem todos os comportamentos.

Segurança, licenças e dados pessoais

Um site é a porta de entrada da empresa na internet. Ele precisa de fechadura.

■ Riscos de segurança mais frequentes

O OWASP Top 10, lista de referência dos riscos mais críticos em aplicações web, inclui categorias como controle de acesso falho, falhas criptográficas, injeção, configurações inseguras, componentes vulneráveis e desatualizados e falhas de identificação e autenticação. Código gerado por IA pode conter qualquer um desses problemas, especialmente quando o pedido não menciona segurança.

Tabela 7.1 — Riscos comuns e cuidados no código gerado por IA

Risco	Exemplo	Cuidado
Controle de acesso falho	Usuário acessa dados de outro mudando um número na URL	Verificar permissões no servidor em toda consulta
Injeção	Dados do formulário montados diretamente em consultas ao banco	Consultas parametrizadas e validação de entradas
Segredos expostos	Chave de API escrita no código do navegador	Segredos apenas no servidor, em variáveis protegidas
Componentes vulneráveis	Biblioteca desatualizada com falha conhecida	Auditoria de dependências e atualização regular
Configuração insegura	Painel administrativo aberto, mensagens de erro detalhadas em produção	Revisão de configuração antes do lançamento
Autenticação frágil	Senhas sem limite de tentativas	Bibliotecas e serviços de autenticação confiáveis

■ Licenças de software

Todo projeto moderno usa bibliotecas de código aberto, cada uma com sua licença. Licenças permissivas, como MIT e Apache, permitem uso comercial com poucas obrigações, como manter o aviso de direitos. Licenças com obrigações de compartilhamento, como a família GPL, podem exigir que o código derivado seja distribuído sob a mesma licença em certas situações. Antes de adotar uma biblioteca, verifique a licença; ferramentas automáticas listam as licenças de todas as dependências do projeto.

O código gerado por IA também merece atenção: em casos raros, ele pode reproduzir trechos de código existente. Revisões, ferramentas de verificação e a preferência por soluções simples e próprias reduzem esse risco. A proteção do software no Brasil é regida pela Lei nº 9.609/1998, e o contrato entre quem desenvolve e quem contrata deve deixar claro a quem pertence o código entregue.

■ Dados pessoais em sites e aplicativos

Formulários de contato, cadastros, agendamentos e ferramentas de análise tratam dados pessoais. A LGPD exige transparência, base legal, segurança e respeito aos direitos dos titulares. Na prática:

- colete apenas os dados necessários para a finalidade do formulário;
- publique uma política de privacidade clara, que explique o que é coletado, para quê, por quanto tempo e com quem é compartilhado;
- peça consentimento para cookies e rastreamentos que não sejam essenciais;
- proteja os dados em trânsito e em repouso e restrinja quem pode acessá-los;
- tenha um canal para que as pessoas exerçam seus direitos, como acesso, correção e exclusão.

■ Hospedagem, cópias e monitoramento

- hospedagem com certificado de segurança, atualizações aplicadas e cópias de segurança automáticas;
- monitoramento de disponibilidade e de erros, com alertas;
- registro de quem tem acesso aos painéis de hospedagem, domínio e banco de dados;
- domínio registrado em nome da empresa, e não de um fornecedor ou funcionário.

! ATENÇÃO

Quem é dono do quê

Antes de contratar ou entregar um site ou aplicativo, defina por escrito: em nome de quem estão o domínio, a hospedagem, as contas de serviços, o repositório de código e as credenciais. Muitas empresas descobrem, na troca de fornecedor, que não têm acesso ao próprio site.

■ Checklist de lançamento



CHECKLIST

Antes de publicar um site ou aplicativo

- ☐ Conteúdo revisado, sem textos de exemplo, números ou depoimentos inventados.
- ☐ Formulários testados de ponta a ponta, com mensagens de sucesso e de erro.
- ☐ Política de privacidade publicada e coerente com o que o site coleta.
- ☐ Consentimento para cookies não essenciais configurado.
- ☐ Auditoria de acessibilidade feita, com correção dos problemas graves.
- ☐ Core Web Vitals medidas na página inicial e nas páginas principais.
- ☐ Títulos, descrições e mapa do site configurados.
- ☐ Dependências auditadas e sem vulnerabilidades graves conhecidas.
- ☐ Segredos fora do código e acessos administrativos protegidos.
- ☐ Cópias de segurança automáticas e monitoramento de erros ativos.
- ☐ Domínio, hospedagem e repositório em nome da empresa.
- ☐ Página de erro amigável e redirecionamentos de endereços antigos.

Implantação de um projeto e indicadores

■ Um projeto típico em fases

Para um site institucional com algumas integrações, como formulário, agenda e CRM, um cronograma de cerca de oito semanas é uma referência razoável quando o conteúdo está disponível e as decisões são tomadas no tempo combinado. Projetos com sistemas sob medida, aplicativos ou muitas integrações pedem mais tempo. O exemplo abaixo mostra a distribuição típica das fases.



Figura 8.1 — Fases de um projeto de site institucional com apoio de IA

Os prazos variam muito conforme o escopo. O ponto central é que cada fase tem um produto verificável, e que a IA encurta as fases de produção, não as de decisão e verificação.

■ Indicadores

Tabela 8.1 — Indicadores para sites e aplicativos

Indicador	Como medir	Leitura
Core Web Vitals	Ferramentas de medição de campo e de laboratório	Experiência de carregamento e interação
Problemas de acessibilidade	Auditoria automática e testes manuais	Inclusão e conformidade
Taxa de conversão	Ações realizadas ÷ visitas	Cumprimento do objetivo do site
Erros em produção	Erros registrados por semana	Estabilidade
Disponibilidade	Tempo no ar ÷ tempo total	Confiabilidade da hospedagem
Tempo de entrega de mudanças	Dias entre o pedido e a publicação	Agilidade da manutenção
Vulnerabilidades em dependências	Auditoria periódica	Segurança

Estudo de caso ilustrativo e exercícios



ESTUDO DE CASO ILUSTRATIVO

Clínica Sorriso Pleno: do site pronto em minutos ao site que agenda

A Clínica Sorriso Pleno, de odontologia, contratou um site gerado rapidamente com IA. Em poucos dias, tinha uma página bonita, com gradientes, três cartões de serviços e depoimentos de pacientes que ninguém havia escrito. O site recebia visitas, mas quase ninguém agendava, e a recepção continuava atendendo tudo por telefone.

Descoberta. A nova equipe começou por entrevistas com a recepção e com pacientes. A principal dúvida dos visitantes era se a clínica atendia determinado convênio e quais eram os horários disponíveis. Nada disso estava no site.

Direção de design. Foram escolhidas referências de sites de saúde com tom acolhedor e informações claras. O sistema visual usou as cores da clínica, tipografia legível e fotografias reais do consultório e da equipe, com autorização. Os depoimentos falsos foram removidos.

Protótipo e testes. O protótipo, gerado com apoio de IA a partir de esboços, foi testado com cinco pacientes. Três não encontraram o botão de agendamento no celular; ele foi movido e destacado.

Desenvolvimento. O agendamento on-line foi integrado à agenda da clínica, com confirmação por mensagem. Os formulários passaram a coletar apenas nome, telefone e convênio, com política de privacidade e consentimento para comunicações.

Qualidade. Antes do lançamento, a equipe corrigiu problemas de contraste e de navegação por teclado, reduziu o tamanho das imagens e configurou cópias de segurança e alertas de erro. O domínio, que estava em nome do primeiro fornecedor, foi transferido para a clínica.

Acompanhamento. A clínica passou a medir agendamentos on-line, ligações para a recepção e o desempenho do site no celular.

Caso ilustrativo: empresa, pessoas e números são fictícios, criados para fins didáticos. Não representam clientes nem resultados da D.B.



EXERCÍCIO DE APLICAÇÃO

Exercício 1: diagnóstico do site atual

1. Abra o site da sua empresa no celular e anote as três primeiras dificuldades.
2. Verifique quais sinais de interface genérica da tabela do capítulo 1 aparecem nele.
3. Liste as perguntas mais frequentes de clientes e confira se o site as responde.



EXERCÍCIO DE APLICAÇÃO

Exercício 2: briefing de design

1. Escolha cinco referências de sites e registre o que funciona em cada uma.
2. Defina os tokens principais do sistema visual da sua marca.
3. Escreva o briefing de design para o assistente com o checklist do capítulo 3.



EXERCÍCIO DE APLICAÇÃO

Exercício 3: qualidade e propriedade

1. Meça as Core Web Vitals da página inicial do seu site.
2. Teste a navegação apenas pelo teclado e registre os problemas.
3. Verifique em nome de quem estão o domínio, a hospedagem e o código.

Como a D.B. pode atuar e conclusão

■ Conclusão

A inteligência artificial reduziu drasticamente o tempo entre a ideia e a primeira versão de um site ou aplicativo. Isso é uma excelente notícia, desde que a velocidade seja usada para explorar mais alternativas, validar com pessoas reais e investir mais tempo onde ele importa: direção de design, qualidade, acessibilidade, desempenho e segurança.

Um produto digital bem feito com IA tem a cara da empresa, funciona no celular de qualquer cliente, é acessível, carrega rápido, protege os dados das pessoas e pode ser mantido por anos. Esse resultado depende de processo, e o processo continua sendo uma responsabilidade humana.

D COMO A D.B. PODE AJUDAR

Como a D.B. desenvolve sites e sistemas com IA

A D.B. Soluções Digitais desenvolve sites, sistemas e aplicativos com apoio de IA e processo profissional:

- descoberta, arquitetura de informação e direção de design com identidade própria;
- protótipos validados com usuários antes do desenvolvimento;
- código revisado, versionado e testado, com documentação;
- acessibilidade, desempenho e segurança como critérios de entrega;
- domínio, hospedagem e código em nome do cliente, com transferência de conhecimento.

Glossário

Acessibilidade

Qualidade de um produto digital que pode ser usado por pessoas com diferentes capacidades, inclusive com deficiência.

CLS

Métrica de estabilidade visual que mede deslocamentos inesperados de conteúdo durante o carregamento.

Core Web Vitals

Conjunto de métricas do Google sobre carregamento, responsividade e estabilidade de páginas.

Dívida técnica

Custo futuro gerado por soluções rápidas ou provisórias no código.

INP

Métrica que mede o tempo de resposta de uma página às interações do usuário.

LCP

Métrica que mede o tempo até o maior elemento visível da página ser exibido.

Pirâmide de testes

Modelo que recomenda muitos testes rápidos de unidade e menos testes lentos de ponta a ponta.

Protótipo

Versão simplificada de um produto usada para validar ideias antes do desenvolvimento.

Sistema visual

Conjunto de decisões de cor, tipografia, espaço e componentes de um projeto.

Tokens de design

Variáveis nomeadas que armazenam as decisões do sistema visual.

WCAG

Diretrizes internacionais de acessibilidade para conteúdo web, publicadas pelo W3C.

REFERÊNCIAS

Referências

As referências abaixo fundamentam conceitos, normas e boas práticas citados no texto. A organização temática da coleção tomou como ponto de partida a biblioteca aberta Altari AI Guide (altari.ai/guide); nenhum texto, imagem ou diagrama dela foi reproduzido. Ferramentas e produtos de terceiros mudam com frequência: confira a documentação oficial antes de implementar. Links verificados em outubro de 2026.

1. ALTARI. *Websites That Look Designed, Screenshot to Code, Reverse Engineer Anything, The \$500k App Stack e Instant Architecture Maps*. Altari AI Guide, 2026. Disponível em: <https://altari.ai/guide>. Referência temática; acesso em: 21 out. 2026.
2. ABI. *screenshot-to-code*. Repositório de código aberto. Disponível em: <https://github.com/abi/screenshot-to-code>.
3. BRASIL. Lei nº 9.609, de 19 de fevereiro de 1998. Lei de Software. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/l9609.htm.
4. BRASIL. Lei nº 13.146, de 6 de julho de 2015. Lei Brasileira de Inclusão da Pessoa com Deficiência. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm.
5. BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm.
6. COHN, Mike. *Succeeding with agile: software development using Scrum*. Boston: Addison-Wesley, 2009.
7. FROST, Brad. *Atomic design*. 2016. Disponível em: <https://atomicdesign.bradfrost.com/>.
8. GOOGLE. *Web Vitals*. web.dev. Disponível em: <https://web.dev/articles/vitals>.
9. KRUG, Steve. *Não me faça pensar: atualizado*. Rio de Janeiro: Alta Books, 2014.
10. LUPTON, Ellen. *Pensar com tipos: guia para designers, escritores, editores e estudantes*. São Paulo: Cosac Naify, 2006.
11. NIELSEN, Jakob. *10 usability heuristics for user interface design*. Nielsen Norman Group, 1994, atualizado em 2024. Disponível em: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
12. OPEN SOURCE INITIATIVE. *Licenses*. Disponível em: <https://opensource.org/licenses>.
13. OWASP FOUNDATION. *OWASP Top 10: 2021*. Disponível em: <https://owasp.org/Top10/>.
14. W3C. *Web Content Accessibility Guidelines (WCAG) 2.2*. 2023. Disponível em: <https://www.w3.org/TR/WCAG22/>.

— PRÓXIMO PASSO

Da leitura à *implementação*.

A D.B. desenvolve sites e sistemas com apoio de IA e com processo profissional: direção de design, código revisado, testes, acessibilidade e desempenho.

Diagnóstico

Mapeamos processos, dados e ferramentas para encontrar onde a IA e a automação fazem sentido.

Automações e integrações

Conectamos sistemas e criamos fluxos com regras claras, tratamento de falhas e registro.

IA aplicada

Assistentes, agentes e bases de conhecimento com limites definidos e revisão humana.

Sistemas e sites

Desenvolvimento sob medida, com foco em uso real, desempenho e evolução.

Conversa inicial sem compromisso: você explica o contexto, nós ajudamos a enxergar o caminho.

D.B. Soluções Digitais

Além do código.

WhatsApp (98) 99107-0234

Conversar com a D.B. ↗