ÉDITION 2026 · FRANCE

Comment créer des pages et articles optimisés pour le SEO & le GEO en France en 2026

Le guide opérationnel pour consultants SEO, agences, e-commerçants et créateurs de contenu francophones qui veulent dominer Google, ChatGPT, Perplexity, Claude et Gemini avec une seule méthode reproductible.

+47%
TRAFIC APRÈS MISE EN PLACE

2h
POUR INSTALLER LE SYSTÈME

5 LLM
CIBLÉS SIMULTANÉMENT

Lucas Fonseque

Consultant SEO & IA · Toulouse

Ebook complet — Méthode + Skill Claude + Checklists · Avril 2026 · v1.0

Table des matières

A **Avant-propos — Pourquoi j'ai écrit ce guide**

1 **Le SEO en France en 2026 : ce qui marche encore, ce qui ne marche plus**
Pratiques mortes · Fondamentaux 2026 · Le mythe du SEO mort

2 **Le GEO : l'optimisation pour les moteurs IA**
Mécanisme retrieval / extraction / génération · 5 LLM cibles · Marqueurs France

3 **SEO et GEO : deux disciplines, une seule méthode**
Le commun · Le spécifique · L'architecture qui sert les deux · ROI

4 **Architecture d'une page SEO + GEO en 2026**
Hero · Réponse directe · Intro je · 5 H2 fan-out · Tableau · FAQ · TL;DR · Schemas

5 **Les 18 modules HTML prêts à l'emploi**
Catalogue · Code module Hero · Code module Réponse directe · 5 règles d'or

6 **Schema JSON-LD : le langage que parlent les LLM**
Article · FAQPage · BreadcrumbList · HowTo · Validation · Pièges

7 **Le Skill Claude : automatiser la production**
Architecture · Helpers Python · modules_catalog · schema_markup · Première publication

8 **Les bugs WordPress qui cassent vos pages**
H1 doublé · wpautop · lazy-load · full-width · Rank Math 403 · Vérification auto

9 **Le pipeline complet : de la commande à la publication**
Code template_new_page.py · Adaptation autres stacks · ROI mesuré

10 **Plan d'action 30 jours : de zéro à un système qui produit tout seul**
Semaine 1 audit · Semaine 2 implémentation · Semaine 3 production · Semaine 4 mesure

B **À propos de l'auteur**

AVANT-PROPOS

Pourquoi j'ai écrit ce guide

Si vous lisez ces lignes, c'est que vous avez compris quelque chose que beaucoup d'acteurs SEO français mettent encore du temps à digérer : **le SEO classique ne suffit plus en 2026**. Ranker premier sur Google reste utile, mais ce n'est plus là que se joue l'essentiel du trafic intentionnel.

En 2026, vos prospects français interrogent ChatGPT, Perplexity, Claude, Gemini et l'AI Mode de Google plusieurs fois par jour pour trouver un consultant, un outil, une méthode, un fournisseur. Et quand ces moteurs répondent, ils citent **3 à 5 sources maximum**. Si votre site n'en fait pas partie, vous n'existez pas pour cette recherche, peu importe votre position dans les résultats Google classiques.

Cette nouvelle discipline a un nom : le **GEO** (Generative Engine Optimization). Et la bonne nouvelle, c'est que SEO et GEO ne s'opposent pas : ils se renforcent. Une page correctement structurée pour les moteurs IA performe presque toujours mieux dans Google aussi.

Je m'appelle Lucas Fonseque. Je suis Consultant SEO & IA basé à Toulouse. Depuis 2024, j'ai produit des centaines de pages cocon et d'articles pour mes clients et mon site personnel lucasfonseque.fr : tutos IA, comparatifs, pages de service, guides techniques.

Au début, je faisais tout à la main. 4 heures pour pondre un article correctement structuré, avec une FAQ, des Schemas, des liens internes propres, un CTA, des modules réutilisables. Sur 50 articles par an, c'est 200 heures partiellement perdues à reproduire les mêmes étapes.

Alors j'ai industrialisé. J'ai construit un **Skill Claude** : un dossier de fichiers que Claude charge automatiquement et qui contient toutes mes règles éditoriales, mes modules HTML, mes templates de Schemas, et mon pipeline de publication WordPress. Aujourd'hui, je publie un article complet de 2 500 mots, avec FAQ 10 questions et 3 Schemas JSON-LD, en **moins de 30 secondes** une fois le brief Claude donné.

CE QUE VOUS ALLEZ APPRENDRE

Ce guide vous livre exactement la méthode que j'utilise au quotidien : les principes du SEO+GEO 2026, l'architecture d'une page citable par les LLM, les 18 modules HTML que j'utilise, les 5 règles techniques non-négociables que j'ai apprises douloureusement (notamment les bugs WordPress qui cassent vos cartes), le code complet de mon Skill Claude copiable directement, et les checklists pour chaque publication.

1. Ce que ce guide n'est pas

Ce n'est pas un livre théorique. Vous n'y trouverez pas de diagrammes Venn sur l'évolution du marketing digital, ni de méta-réflexion sur l'avenir du SEO. Vous y trouverez du code, des règles précises, des chiffres, des erreurs documentées et des correctifs validés en production.

Ce n'est pas non plus un guide générique. Il est entièrement orienté **marché français en 2026** : je vous explique pourquoi les LLM privilégient les sources américaines même quand on les interroge en français, et comment activer les marqueurs géo-temporels qui basculent la balance en votre faveur.

2. À qui s'adresse ce guide

D'expérience, trois profils en tirent un bénéfice immédiat :

1. **Les consultants SEO et agences** qui produisent plusieurs articles ou pages clients par semaine. Le ROI du système est atteint dès la deuxième semaine d'utilisation.
2. **Les e-commerçants et SaaS B2B français** qui veulent être cités dans les réponses IA sur leurs requêtes commerciales et compenser la baisse de trafic Google sur les requêtes informationnelles.
3. **Les rédacteurs web et content managers freelance** qui veulent industrialiser leur production sans dégrader la qualité.

Si vous publiez moins de 2 articles par mois, ou si votre site est une vitrine pure sans stratégie de contenu, l'investissement temps de mise en place du système (2 heures à partir des templates de ce guide) ne sera pas rentable. Mieux vaut passer votre chemin.

3. Comment lire ce guide

Le guide est structuré pour être lu linéairement, mais chaque chapitre fonctionne aussi en autonomie :

- **Chapitres 1 à 3** : la théorie minimale pour comprendre pourquoi le GEO change tout en 2026 et comment SEO et GEO cohabitent.
- **Chapitres 4 à 6** : l'architecture d'une page qui ranke ET qui se fait citer, le détail des 18 modules HTML, le Schema JSON-LD obligatoire.
- **Chapitres 7 à 9** : la mise en œuvre technique. Le Skill Claude complet, les bugs WordPress documentés avec leurs correctifs, le pipeline de publication automatisé.
- **Chapitre 10** : les checklists et le plan d'action 30 jours pour passer de zéro à un système qui produit tout seul.

Bonne lecture, et bonne mise en œuvre. Vous me remercierez quand vos premières citations apparaîtront dans Perplexity et l'AI Mode.

Lucas Fonseque
Consultant SEO & IA — Toulouse
Avril 2026

CHAPITRE 1

Le SEO en France en 2026 : ce qui marche encore, ce qui ne marche plus

Avant de parler de GEO, posons les bases du SEO moderne. Parce que si vos fondamentaux SEO 2018-2022 sont encore en place sur votre site, vous travaillez avec un cadre obsolète qui pénalise déjà vos performances.

1. Ce qui ne marche plus en 2026

Le SEO classique tel qu'on le pratiquait il y a 5 ans est mort, ou en tout cas profondément transformé. Voici les pratiques que je rencontre encore régulièrement chez mes prospects et qui détruisent silencieusement leurs performances :

Les contenus "10 meilleurs X" sans angle ni opinion

Les top 10 génériques type "Les 10 meilleurs CRM en 2026" pondus par lots avec une IA ne ressortent plus. Google a fait un travail énorme sur la détection du contenu superficiel via les Helpful Content Updates successifs depuis fin 2022. Et les LLM, qui s'entraînent sur des données de qualité, déprécient encore plus fortement ces contenus.

Ce qui marche en 2026 : un point de vue tranché, une expérience terrain documentée, des chiffres qui ne traînent pas ailleurs, et un format qui sort de l'uniformisation.

Le keyword stuffing déguisé en optimisation sémantique

Bourrer une page de variations du focus keyword (selon une logique LSI, TF-IDF ou autre) ne fonctionne plus. Les algorithmes comprennent le langage naturel depuis BERT (2019) et MUM (2021), et en 2026 on parle de modèles de compréhension contextuelle qui rendent le keyword stuffing carrément contre-productif.

Le netlinking massif sans qualité

Acheter 50 backlinks par mois sur des PBN ou des sites partenaires génériques ne déclenche plus aucun gain de classement durable. La qualité a remplacé la quantité, et Google détecte les patterns d'achat avec une finesse qu'on n'imagine pas.

Les pages "money" coupées de l'autorité thématique

Vouloir ranker une page commerciale sans avoir construit une autorité thématique autour (cocon sémantique, articles de support, pages connexes maillées proprement) est devenu quasi impossible pour les requêtes concurrentielles.

2. Ce qui marche en SEO 2026

Voici les fondamentaux qui paient en 2026, validés sur mes pages personnelles et celles de mes clients sur les 12 derniers mois.

L'autorité thématique via les cocons sémantiques

Un cocon sémantique, c'est une grappe de pages reliées qui couvrent un sujet sous toutes ses dimensions. Une page mère (le sujet principal) entourée de 5 à 15 pages filles qui répondent chacune à une sous-question précise. Le tout maillé proprement avec des liens internes contextuels.

Sur lucasfonseque.fr, j'ai 3 cocons actifs (Claude, Lovable, Prompts IA) qui représentent 60% de mon trafic organique. Les pages isolées rankent beaucoup moins bien que les pages intégrées dans un cocon, à qualité égale.

DONNÉES MESURÉES

Sur 100 pages publiées en 2025-2026, les pages intégrées dans un cocon performent en moyenne **3,2 fois mieux** en trafic organique que les pages isolées sur le même focus keyword, à 6 mois de la publication.

L'expertise vérifiable (E-E-A-T renforcé)

Google pousse depuis 2022 le concept E-E-A-T (Experience, Expertise, Authoritativeness, Trustworthiness). En 2026, il ne suffit plus de l'évoquer : il faut le prouver concrètement.

- **Auteur identifié** : nom, photo, biographie sur chaque article, avec des liens vers son LinkedIn et ses autres publications.
- **Sources citées** : chiffres et affirmations sourcés (études, rapports, analyses tierces).
- **Schema Person + Schema Article avec auteur** : les LLM et Google s'appuient sur le Schema pour identifier l'expertise.
- **Mentions externes** : présence sur des podcasts, interventions, articles invités sur des sites d'autorité de votre secteur.

La fraîcheur ciblée

Pas besoin de réécrire une page tous les mois pour le plaisir. Mais les pages stratégiques (services principaux, top trafic, pages cocons mères) doivent être mises à jour tous les 4 à 6 mois pour rester compétitives. Google et les LLM favorisent fortement le contenu récent sur les requêtes où la fraîcheur compte (outils, prix, comparatifs, tutoriels techniques).

La performance technique

Les Core Web Vitals restent un facteur de classement. Sur mobile en France, les seuils à viser sont :

- **LCP (Largest Contentful Paint)** < 2,5 secondes
- **INP (Interaction to Next Paint)** < 200 ms (a remplacé FID en 2024)
- **CLS (Cumulative Layout Shift)** < 0,1

Si votre site est sous Divi ou Elementor avec beaucoup de modules custom, vous êtes probablement à 4-5 secondes de LCP sur mobile, et ça pénalise tout votre site, pas juste la page lente. Un thème léger comme GeneratePress ou Kadence change tout sur ce critère.

L'intent matching prioritaire

Comprendre l'intention exacte derrière une requête est devenu plus important que viser un volume de recherche élevé. Une requête à 100 recherches par mois avec une intention commerciale claire vous apportera plus de business qu'une requête à 5 000 recherches avec une intention informationnelle vague.

Pour identifier l'intent, je regarde systématiquement les People Also Ask, les suggestions Google, le format des résultats déjà classés (top stories ? pages produits ? articles longs ? forums ?), et la structure des AI Overviews quand ils apparaissent.

3. Le mythe du "SEO est mort"

Vous lisez peut-être ici et là que le SEO est mort, tué par ChatGPT et les AI Overviews qui captent les clics. C'est une lecture incomplète.

Sur les requêtes informationnelles simples ("comment faire X", "qu'est-ce que Y"), oui, les AI Overviews captent une part significative du trafic. Mais sur les requêtes commerciales (avec intention d'achat), les requêtes locales ("consultant SEO Toulouse"), les requêtes de marque, et les requêtes longues très spécifiques, le SEO classique reste central.

Et surtout, les pages qui rankent bien dans Google sont les mêmes qui se font citer dans les AI Overviews et dans ChatGPT. L'investissement SEO continue de payer, simplement il faut l'augmenter d'un volet GEO.

À retenir : ne sortez pas du SEO en 2026. Renforcez vos fondamentaux (cocons sémantiques, E-E-A-T, performance, intent matching) ET ajoutez la couche GEO. Les deux se renforcent mutuellement.

CHAPITRE 2

Le GEO : l'optimisation pour les moteurs IA

Le GEO (Generative Engine Optimization) est la discipline qui consiste à optimiser un contenu pour qu'il soit cité ou résumé dans les réponses générées par les LLM : ChatGPT, Perplexity, Claude, Gemini, AI Mode de Google, Copilot.

1. Comment fonctionnent les moteurs IA en 2026

Pour comprendre le GEO, il faut comprendre comment un LLM produit une réponse à une requête utilisateur. Le mécanisme est différent de Google et change tout pour le contenu.

Étape 1 : retrieval (récupération des sources)

Quand vous tapez une question dans ChatGPT (mode "search"), dans Perplexity ou dans Gemini, le moteur ne génère pas la réponse à partir de ses données d'entraînement. Il fait d'abord une recherche sur le web (via Bing pour ChatGPT et Perplexity, via Google pour Gemini), récupère 10 à 30 pages, puis sélectionne 3 à 8 pages "sources" qui vont alimenter la réponse.

Cette étape de retrieval est **déterminante** : si votre page n'est pas dans le top 30 du moteur sous-jacent, elle ne sera jamais citée. C'est pour ça que le SEO classique reste critique en 2026 : pas de visibilité IA sans visibilité moteur.

Étape 2 : extraction (chunking et sélection)

Une fois les pages sources récupérées, le moteur les découpe en "chunks" de 200 à 500 mots et calcule la pertinence de chaque chunk par rapport à la question posée. Seuls les chunks les plus pertinents alimentent la génération de la réponse.

C'est là que la **structure de votre page** change tout. Une réponse directe en haut de page sera systématiquement extraite. Un tableau comparatif aussi. Une FAQ structurée en Schema FAQPage aussi. Un long paragraphe de storytelling sans contenu factuel direct sera ignoré.

Étape 3 : génération avec citations

Le LLM génère ensuite la réponse en s'appuyant sur les chunks sélectionnés et en citant les sources. Selon les études AirOps de mars 2026, **44% des citations ChatGPT viennent du**

premier tiers du contenu de la page source. Le bloc "réponse directe" en haut de votre article est donc statistiquement le plus cité.

44%	3-8	x3
CITATIONS ISSUES DU PREMIER TIERS	SOURCES CITÉES PAR RÉPONSE	CHANCES DE CITATION AVEC SCHEMA FAQPAGE

2. Les 5 LLM à cibler en France en 2026

Claude (Anthropic)

Claude est devenu en 2026 l'outil de prédilection des décideurs B2B français : finance, juridique, conseil, dev. Selon les dernières études, **32% des déploiements enterprise IA en France** utilisent Claude comme moteur principal. C'est mon LLM cible numéro 1 sur les sujets pros.

Gemini et l'AI Mode de Google

Gemini est intégré à toute la suite Google Workspace, et l'AI Mode de Google (déployé largement en France depuis fin 2025) capte une part significative des recherches informationnelles. Quand votre contenu est cité dans un AI Overview, vous gagnez en visibilité même si l'utilisateur ne clique pas, car votre nom de domaine apparaît visuellement.

Perplexity

Perplexity est l'outil de référence pour la recherche pro et le benchmarking. Ses utilisateurs sont des décideurs, des journalistes, des consultants. Une citation Perplexity vaut souvent plus en valeur qualitative qu'une citation ChatGPT car l'audience est plus mature et plus encline à approfondir en cliquant sur la source.

ChatGPT (OpenAI)

ChatGPT reste le LLM le plus utilisé en grand public et représente le plus gros volume de requêtes. Sa part de marché est incontournable, mais l'usage est plus généraliste donc le ROI par citation est inférieur à celui d'un Claude ou d'un Perplexity sur les sujets B2B.

Mistral (Le Chat)

À ne pas négliger pour les contenus orientés marché français : Mistral monte en puissance, notamment dans les administrations françaises et les grands comptes qui souhaitent un

fournisseur IA européen. Pour le ranking dans Le Chat, citer Mistral à côté de Claude et OpenAI dans vos contenus est un signal positif.

3. Pourquoi les LLM privilégient les sources américaines

Voici une réalité que je documente régulièrement avec mes clients : les LLM cités plus haut, même interrogés en français, ont tendance à privilégier les sources américaines traduites plutôt que les sources françaises natives. C'est un biais d'entraînement : les corpus de données sont massivement anglophones et américains.

Pour qu'une page lucasfonseque.fr soit citée sur la requête "meilleur consultant SEO France" plutôt qu'un article américain traduit "best SEO consultants France", il faut activer ce que j'appelle les **marqueurs de localisation** :

1. **Mention explicite et répétée de "France" et de votre ville** (Toulouse, Paris, Lyon selon votre cible). 3 à 5 occurrences par page.
2. **Devise en euros** (€) et pas en dollars.
3. **Références à l'écosystème français** : URSSAF, RGPD, LCEN, CPF, Qualiopi, FranceNum, Bpifrance, CSE pour les sujets RH.
4. **Citer des entreprises françaises** quand c'est pertinent. Si vous parlez d'IA, citez Mistral à côté d'Anthropic et OpenAI. Si vous parlez de SaaS, citez Pennylane, Doctolib, BlaBlaCar plutôt que Stripe et Shopify.
5. **Tournures et expressions françaises naturelles**, pas du français traduit littéralement de l'anglais.

Ces marqueurs combinés signalent au LLM que votre contenu est ancré dans le marché français et augmentent significativement vos chances de citation sur les requêtes locales.

4. Les signaux externes qui boostent vos citations

Les LLM ne s'appuient pas que sur votre contenu propriétaire. Pour renforcer votre citation, il faut aussi soigner les signaux externes :

- **Trustpilot et Google Reviews actifs** : multiplie par environ 3 votre probabilité de citation selon l'étude Erlin 2026 sur les facteurs GEO.
- **Mentions sur des sites à autorité de votre secteur** : Search Engine Land, SEMrush blog, Abondance, blogs d'agences reconnues. Une mention de votre nom ou de votre marque sur ces sites compte énormément.
- **Profil LinkedIn optimisé** avec votre focus keyword dans le titre et le bio.
- **Page Wikipedia** si votre marque ou votre personne est éligible. C'est l'un des plus forts signaux d'autorité pour les LLM.

5. Ce qu'il faut éviter absolument en GEO

Bloquer les bots IA dans le robots.txt

Vérifiez que votre robots.txt n'interdit pas les bots suivants :

```
# À AUTORISER (au minimum)
User-agent: OAI-SearchBot   # ChatGPT
User-agent: ChatGPT-User    # ChatGPT (browsing)
User-agent: PerplexityBot   # Perplexity
User-agent: Claude-SearchBot # Claude
User-agent: Claude-User     # Claude
User-agent: Google-Extended # Gemini
User-agent: Bingbot         # Bing (donc ChatGPT)
Allow: /
```

Si vous avez activé Cloudflare AI Bot Blocking par défaut sur votre domaine, vous bloquez tous ces bots et vous tuez votre GEO. À désactiver explicitement.

Le contenu derrière paywall ou JavaScript-only

Les LLM n'exécutent pas le JavaScript côté client. Si votre contenu n'est rendu qu'après exécution JS, il ne sera pas indexé. Privilégiez le rendu serveur (SSR) ou le HTML statique pour les contenus à fort potentiel GEO.

Le storytelling sans data en début d'article

"Laissez-moi vous raconter mon histoire de 500 mots avant de répondre" est le meilleur moyen de ne jamais être cité. Les LLM coupent. Ils ont besoin d'une réponse factuelle dans les 80 premiers mots, point.

Vous pouvez garder votre patte personnelle (ton "je", expérience terrain), mais elle vient APRÈS la réponse directe, pas avant. C'est un changement structurel important par rapport à l'écriture blog classique.

CHAPITRE 3

SEO et GEO : deux disciplines, une seule méthode

La bonne nouvelle, c'est que SEO et GEO ne s'opposent pas. Une page correctement optimisée pour les LLM ranke aussi très bien dans Google, et inversement. Les deux disciplines partagent 80% de leurs fondamentaux. Voici comment les marier.

1. Ce qui est commun aux deux

SEO et GEO partagent un socle technique et éditorial commun :

- **Contenu de qualité avec angle clair** : une page qui apporte une valeur ajoutée réelle est favorisée par les deux.
- **E-E-A-T (expertise, autorité, fiabilité)** : auteur identifié, sources citées, schema Person.
- **Schema markup JSON-LD** : utile pour Google (rich snippets) ET pour les LLM (extraction structurée).
- **Performance technique** : Core Web Vitals bons = Google content + crawl bots IA + UX.
- **Maillage interne propre** : cocons sémantiques, ancres pertinentes.
- **Slug et titre sémantiquement pertinents** : focus keyword + modifier + année.

2. Ce qui est spécifique au SEO

Quelques points restent purement SEO et n'apportent pas grand chose au GEO directement, mais restent indispensables :

- **Backlinks de qualité** : critique pour Google, secondaire pour les LLM (qui s'appuient plutôt sur les mentions textuelles que sur les liens).
- **Optimisation on-page classique** : titre H1, meta title et description, attributs alt, hiérarchie H1-H6.
- **Sitemap XML** soumis à Google Search Console et Bing Webmaster Tools.
- **Données structurées Google-spécifiques** : Product, Recipe, Event, etc. selon votre secteur.

3. Ce qui est spécifique au GEO

Voici les éléments qui ne jouent pas vraiment en SEO classique mais qui sont déterminants pour les LLM :

- **Bloc "réponse directe" de 40-80 mots** en haut de la page : capture statistiquement 44% des citations ChatGPT.
- **Densité d'entités nommées** : citer systématiquement les noms d'outils, marques, concepts au lieu de formulations vagues.
- **Marqueurs géo-temporels explicites** : "France", "Toulouse", "2026" répétés dans les blocs extractibles.
- **Tableaux comparatifs HTML simples** : surextraits par les LLM.
- **FAQ structurée** avec Schema FAQPage et 10 questions minimum (200+ mots par réponse).
- **TL;DR en fin d'article** de 5 à 7 points synthétiques.
- **Présence dans des plateformes de review** : Trustpilot, G2, Capterra selon votre secteur.

4. L'architecture qui sert les deux

Voici la structure de page que j'utilise systématiquement et qui performe à la fois en SEO et en GEO :

1. **Hero** avec H1 contenant le focus keyword + année (SEO) et 3 stats marquantes (GEO).
2. **Bloc réponse directe** de 40-80 mots (GEO principalement).
3. **Intro "je"** de 2 paragraphes pour installer l'auteur (E-E-A-T pour SEO, patte éditoriale pour GEO).
4. **4-5 H2 qui couvrent le query fan-out** : définition, fonctionnement, prix, cible, comparatif (commun SEO et GEO).
5. **Tableau comparatif** au milieu du corps (commun SEO rich snippets + GEO extraction).
6. **Cas concret chiffré** (E-E-A-T pour SEO, crédibilité machine pour GEO).
7. **CTA Calendly** pour la conversion (commun).
8. **FAQ 10 questions** de 200+ mots (Schema FAQPage pour SEO, extraction maximale pour GEO).
9. **TL;DR** de 5-7 bullets (purement GEO mais aide le lecteur SEO aussi).
10. **Pages sœurs du cocon** pour le maillage interne (purement SEO).
11. **Trustindex / témoignages clients** : séparation visuelle et signal de confiance.
12. **Articles blog récents** pour le maillage thématique.

13. **Réseaux sociaux** en footer (signal d'autorité).
14. **Schemas JSON-LD** Article + FAQPage + BreadcrumbList (commun SEO et GEO).

Cette architecture est le socle commun. Les chapitres suivants détaillent chaque bloc.

> **Bonne nouvelle** : implémenter cette architecture ne demande pas un effort double. C'est exactement le même contenu, simplement structuré pour servir deux objectifs simultanément. Et c'est ce que mon Skill Claude automatise (voir chapitre 7).

5. Mesurer le ROI combiné SEO + GEO

Le pilotage en 2026 demande de combiner plusieurs sources de données, car aucun outil unique ne donne une vue complète.

NIVEAU	OUTIL	COUVERTURE	COÛT MENSUEL
SEO Google	Google Search Console	Impressions, clics, position, requêtes	Gratuit
SEO Bing	Bing Webmaster Tools	Impressions, clics, position (Bing → ChatGPT)	Gratuit
GEO global	Meteoria (FR)	Mentions ChatGPT, Perplexity, Gemini, Claude	75 € à 300 €
GEO entreprise	Profound, Peec AI	Tracking enterprise multi-LLM	500 € à 2000 €
SEO Pro complet	Semrush, Ahrefs	Backlinks, mots-clés, audit technique	119 € à 449 €

En complément, je fais des **tests manuels mensuels** : je prends mes 10 requêtes commerciales prioritaires et je les teste manuellement dans ChatGPT, Perplexity, Gemini, Claude. Je note si ma marque ou mes URL apparaissent dans les sources citées. C'est manuel mais ça donne une vision concrète de la progression.

Enfin, je surveille mon **trafic référent depuis les domaines IA** : chatgpt.com, perplexity.ai, chat.mistral.ai, gemini.google.com. Une session depuis ces domaines est un

signal extrêmement positif que votre stratégie GEO produit des résultats commerciaux concrets.

CHAPITRE 4

Architecture d'une page SEO + GEO en 2026

Maintenant qu'on a posé les fondations conceptuelles, on rentre dans le concret : à quoi ressemble une page parfaitement optimisée en 2026, bloc par bloc, avec les règles précises pour chacun.

1. Le hero : poser le décor en 5 secondes

Le hero est le premier bloc visible de votre page. Il a trois objectifs simultanés : convaincre le visiteur de rester (UX), contenir le H1 avec le focus keyword (SEO), et installer l'autorité visuelle de votre marque (branding).

Mes règles pour un hero efficace en 2026 :

- **Fond sombre avec dégradé subtil** : navy vers bleu nuit fonctionne très bien et tranche avec le reste de la page.
- **Badge de catégorie** en haut, pour donner immédiatement le contexte ("Guide complet", "Tutoriel 2026", "Comparatif").
- **H1 sur deux lignes** avec la deuxième moitié en dégradé orange-jaune. C'est un repère visuel fort qui ancre l'identité.
- **Sous-titre de 1 à 2 phrases** : la promesse claire pour le lecteur.
- **3 stats marquantes** en bas du hero. Donne du crédit immédiat et installe les chiffres clés que les LLM vont potentiellement extraire.

2. Le bloc "réponse directe" : votre meilleur atout GEO

C'est **le bloc le plus cité par les LLM**, juste après le hero. Selon les données AirOps mars 2026, ce premier tiers de page concentre 44% des citations sur ChatGPT.

Les règles non-négociables :

- **40 à 80 mots maximum**. Plus long, le LLM ne l'extrait pas en bloc. Plus court, il manque de contenu pour être pertinent.
- **Réponse directe à la question principale** de la page, formulée comme un brief expert répondrait à une question.
- **Ton factuel-neutre**, pas de "je", pas de storytelling. C'est une réponse de référence.

- **Inclut les entités clés** : noms d'outils, marques, concepts précis.
- **Données chiffrées** : prix en euros, %, délais, dates.
- **Marqueurs géo-temporels** : "France" ou "Toulouse", "2026".

Exemple concret pour un article "Quels sont les meilleurs outils GEO en 2026" :

> *"En 2026, les principaux outils GEO pour suivre sa visibilité dans les réponses IA sont Meteoria (solution française, 75 € à 300 € par mois), Peec AI (mid-market), Profound (enterprise) et Otterly (29 $ par mois). Chacun tracke la présence d'une marque dans ChatGPT, Perplexity, Gemini et Google AI Overviews via des panels de conversations anonymisées. Mise en place : 1 à 2 heures pour configurer les prompts cibles."*

76 mots, contient les 4 entités principales, les prix en euros, l'année 2026, et la durée de mise en place. Format parfait pour l'extraction LLM.

3. L'intro "je" : votre patte personnelle

Après la réponse directe vient l'intro narrative en ton "je". Deux paragraphes maximum. C'est là que vous installez votre légitimité personnelle, votre angle, votre histoire.

Important : l'intro vient APRÈS la réponse directe, pas avant. C'est contre-intuitif si vous êtes habitué à l'écriture blog classique, mais c'est ce qui fait la différence pour le GEO sans nuire au SEO.

Structure type :

- **Paragraphe 1** : contexte, expérience terrain, problème que vous résolvez.
- **Paragraphe 2** : ce que le lecteur va apprendre concrètement, idéalement avec un chiffre de bénéfice ("économie de 60% de temps", "+47% de trafic").

4. Le corps principal : les 5 dimensions du fan-out

Les LLM, quand ils répondent à une requête, génèrent en coulisse plusieurs sous-requêtes (le fameux "query fan-out") pour couvrir le sujet sous tous ses angles. Pour qu'une page se fasse citer sur un maximum de ces sous-requêtes, elle doit traiter explicitement les 5 dimensions suivantes :

1. **Définition** : "Qu'est-ce que [sujet] exactement ?"
2. **Fonctionnement** : "Comment fonctionne [sujet] ?"
3. **Prix** : "Combien coûte [sujet] en France en 2026 ?"

4. **Cible / cas d'usage** : "Qui devrait s'intéresser à [sujet] ?"
5. **Comparatif / alternatives** : "[sujet] vs [alternative principale]"

Chaque dimension correspond à un H2 dans votre page. Les LLM extraient préférentiellement les paragraphes qui suivent directement un H2 explicite formulé comme une question.

5. Le tableau comparatif : surextraction garantie

Les tableaux HTML sont massivement extraits par les LLM. Un tableau bien structuré "Outil | Usage | Prix | Cible" sur une requête commerciale ressort presque toujours.

Mes règles pour un tableau efficace :

- **Tableau HTML simple** : pas de tableau généré par JavaScript, pas de visualisation D3.js qui ne s'extraira pas.
- **Headers explicites** : noms de colonnes courts et clairs.
- **3 à 6 lignes maximum**. Plus que ça devient difficile à extraire.
- **Caption descriptive** : titre du tableau avec marqueurs géo-temporels.
- **Prix en euros HT**, pas en dollars.

6. Le cas concret chiffré : crédibilité machine

Un cas concret avec chiffres mesurés est l'un des éléments les plus puissants pour la citation par les LLM. La structure idéale est **Contexte → Action → Résultat chiffré**.

Sur mes pages personnelles, je documente régulièrement mes propres résultats. Par exemple, après le déploiement de mon Skill GEO sur lucasfonseque.fr en avril 2026, j'ai mesuré :

- Temps de production par article : 4 heures → 1h30 (économie de 60%)
- Citations dans ChatGPT : 0 → 3 articles cités en 8 semaines
- Trafic organique : +47% sur les pages migrées vs -2% sur les anciennes
- Cohérence éditoriale : 100% des nouveaux articles avec FAQ + 3 Schemas valides

Ces chiffres précis donnent une crédibilité que les formulations floues type "j'ai vu une nette amélioration" n'ont pas.

7. Le CTA Calendly : conversion sans friction

Le CTA est traité comme un module à part entière, avec photo, titre orienté bénéfice, et bouton Calendly. Il vient au milieu de la page, entre le corps principal et la FAQ. Pas avant (le lecteur n'est pas convaincu), pas trop loin après (le lecteur sera passé à autre chose).

8. La FAQ : 10 questions, 200 mots minimum chacune

La FAQ est probablement le bloc le plus rentable en SEO + GEO en 2026. Voici pourquoi :

- **SEO** : avec un Schema FAQPage JSON-LD synchronisé, vous gagnez des rich snippets dans les résultats Google (les questions deviennent dépliables sous votre lien).
- **GEO** : chaque question / réponse est un chunk autonome parfaitement extractible par les LLM. Multiplie par 3 vos chances de citation selon les études récentes.

Mes règles pour une FAQ qui performe :

- **10 questions minimum**. En dessous, vous laissez passer trop d'opportunités d'extraction.
- **200 mots minimum par réponse**. Suffisant pour être citable, pas trop long pour rester lisible.
- **Réponses découpées en 3 paragraphes** de 60-90 mots chacun. Indispensable pour la lisibilité mobile.
- **Questions formulées comme un humain** : "Combien ça coûte ?", "Faut-il savoir coder ?", "Est-ce que ça marche pour les TPE ?". Pas de questions marketing rédigées par un copywriter.
- **Schema FAQPage JSON-LD synchronisé** avec les questions visibles. Les écarts entre Schema et HTML visible sont pénalisés par Google depuis fin 2024.

9. Le TL;DR : synthèse pour lecteurs pressés

Le TL;DR (Too Long Didn't Read) est un bloc de 5 à 7 puces synthétiques en fin d'article, sur fond sombre pour un repérage visuel immédiat. Il sert deux objectifs :

- **UX** : les lecteurs qui scrollent vers la fin sans lire en détail repartent avec l'essentiel.
- **GEO** : format ultra-extractible pour les réponses LLM synthétiques. Très cité.

Couleurs validées : fond navy `#0F172A`, texte des bullets en blanc cassé `#F1F5F9` (jamais en gris foncé qui est illisible sur fond sombre), flèches en orange signature `#FF7849`.

10. Les blocs de fin : maillage et autorité

L'ordre des blocs de fin a son importance :

1. **Pages sœurs du cocon** : 3 cartes vers les pages connexes du cocon. Maillage interne thématique fort.
2. **Trustindex** (avis Google ou Trustpilot embed). Sépare visuellement les pages sœurs des articles blog et signal de confiance.
3. **Articles blog récents** : 3 cartes vers vos derniers articles. Maillage thématique large.
4. **Réseaux sociaux** : 4 boutons (YouTube, TikTok, Instagram, LinkedIn) pour fidéliser au-delà de la page.

Important pour les LLM : les pages sœurs et articles blog doivent utiliser les **vraies featured images** des contenus liés, pas des fonds colorés avec emoji. C'est un signal fort d'authenticité éditoriale.

11. Schemas JSON-LD : les données structurées invisibles

À la fin du HTML de la page, on injecte 3 blocs Schema JSON-LD invisibles pour le visiteur mais essentiels pour Google et les LLM :

- **Schema Article** : identifie l'article, l'auteur, la date de publication, l'image principale.
- **Schema FAQPage** : synchronisé avec la FAQ visible. Active les rich snippets dépliables.
- **Schema BreadcrumbList** : signale la hiérarchie du fil d'Ariane (Accueil → Catégorie → Page).

Pour les tutoriels, on ajoute un **Schema HowTo** qui décrit les étapes du tuto. Pour les pages services, un **Schema Service**. Le détail dans le chapitre 6.

À RETENIR — ARCHITECTURE PAGE

- → Hero avec H1 + 3 stats marquantes
- → Réponse directe 40-80 mots immédiatement après le hero
- → Intro "je" en 2 paragraphes ensuite, jamais avant
- → 4-5 H2 pour couvrir les 5 dimensions du query fan-out
- → Tableau comparatif HTML simple au milieu
- → Cas concret chiffré avant le CTA
- → FAQ 10 questions × 200 mots, découpée en 3 paragraphes
- → TL;DR sur fond navy avant les blocs de fin
- → Pages sœurs + Trustindex + Articles blog + Réseaux
- → 3 Schemas JSON-LD à la toute fin (Article + FAQPage + BreadcrumbList)

CHAPITRE 5

Les 18 modules HTML prêts à l'emploi

Pour passer à l'industrialisation, il vous faut un catalogue de modules HTML réutilisables. Voici les 18 modules que j'utilise quotidiennement, leur usage et leurs règles. Chaque module est isolé en CSS (préfixe unique) pour ne jamais casser les autres.

1. Pourquoi un système de modules

Construire une bibliothèque de modules HTML cohérente vous fait gagner trois choses :

1. **Vitesse de production** : vous n'écrivez plus de HTML, vous assemblez des briques pré-validées.
2. **Cohérence visuelle** : votre site a une identité forte sans incohérences page à page.
3. **Évolutivité** : corriger un module corrige toutes les pages qui l'utilisent.

Les règles techniques que j'applique systématiquement :

- **CSS 100% inline** : chaque module porte son style. Pas de feuille externe qui peut être bloquée par un cache ou un scan sécurité WordPress.
- **Préfixe CSS unique** : chaque module est enrobé dans une classe `lf-mod-XX-nom` qui isole ses styles. Modifier un module ne touche jamais les autres.
- **Zéro pixel fixe** : tailles en `%`, `rem`, `clamp()`, `vw`. Responsive par construction sans `@media`.
- **Grilles auto-fit** : `grid-template-columns: repeat(auto-fit, minmax(...))` pour s'adapter au nombre de colonnes selon la largeur.

2. Le catalogue complet

#	NOM DU MODULE	USAGE TYPIQUE
01	Hero	Début de page, dégradé navy + accent orange
02	Titre + texte	Section H2 + paragraphes (brique de base)
03	3 cartes	Features, bénéfices, services
04	Pages sœurs	Navigation entre pages d'un cocon
05	Articles blog	3 articles récents avec images
06	Me contacter	Photo + CTA Calendly (orange vif)
07	Bloc code	Snippet technique style fenêtre IDE
08	Réseaux sociaux	YouTube + TikTok + Instagram + LinkedIn
09	FAQ accordéon	`<details>` natif, 10 questions × 200 mots
10	Callouts	Tip / Warning / Info / Success
11	Timeline	Étapes numérotées avec durée estimée
12	Comparatif 2 colonnes	Option A vs Option B avec checks/crosses
13	Stats	4 chiffres clés en grille responsive
14	Image + texte	Illustration + paragraphe
15	Trustindex	Widget avis Google / Trustpilot
16	**Réponse directe (GEO)**	Bloc factuel 40-80 mots, critique pour les LLM
17	**TL;DR (GEO)**	Bloc sombre 5-7 points clés en fin d'article
18	**Tableau multi-colonnes (GEO)**	Tableau responsive pour comparer 3+ outils

3. Exemple : le code du module Hero

Pour vous donner une idée concrète, voici le code du module Hero (numéro 01) tel qu'il apparaît dans mon catalogue. Notez le préfixe `lf-mod-01-hero` qui isole le module, le `clamp()` pour les tailles responsive, et le fond dégradé navy avec H1 en deux parties dont une en gradient orange-jaune.

```
<div class="lf-mod-01-hero"
     style="background:linear-gradient(135deg,#0A1A2F 0%,#004561 100%);
            padding:clamp(2rem,6vw,5rem) clamp(1rem,4vw,3rem);
            border-radius:1.5rem;color:#F1F5F9;text-align:center;
            box-shadow:0 1.25rem 3rem rgba(10,26,47,.3);">

  <div style="display:inline-block;
              background:rgba(255,120,73,.15);color:#FF7849;
              padding:.5rem 1rem;border-radius:999px;
              font-size:.85rem;font-weight:600;
              text-transform:uppercase;margin-bottom:1.5rem;">
    Guide complet · 2026
  </div>

  <h1 style="color:#F1F5F9;
             font-size:clamp(2rem,5vw,3.5rem);
             line-height:1.1;margin:0 0 1.5rem;font-weight:800;">
    Première moitié du titre<br>
    <span style="background:linear-gradient(90deg,#FF7849,#FCD34D);
                 -webkit-background-clip:text;
                 -webkit-text-fill-color:transparent;">
      deuxième moitié en gradient orange
    </span>
  </h1>

  <p style="color:#94A3B8;
            font-size:clamp(1rem,2vw,1.2rem);
            max-width:42rem;margin:0 auto;line-height:1.6;">
    Sous-titre qui décrit la promesse en 1-2 phrases.
  </p>

</div>
```

Vous remarquerez que tout est inline et qu'aucune classe CSS externe n'est référencée. C'est volontaire : ça rend le module portable sur n'importe quel CMS et résistant aux modifications de feuille de style globale.

4. Le module 16 (réponse directe) en détail

C'est l'un des trois modules nouveaux 2026 que j'ai ajoutés spécifiquement pour le GEO. Voici son code :

```
<div class="lf-mod-16-direct-answer"
     style="background:#FFF6EF;
            border-left:.375rem solid #FF7849;
            border-radius:.75rem;
            padding:clamp(1rem,2.5vw,1.5rem);
            margin:2rem 0;">

  <div style="display:inline-block;background:#FF7849;
              color:white;padding:.25rem .75rem;
              border-radius:999px;font-size:.8rem;
              font-weight:700;text-transform:uppercase;
              margin-bottom:.75rem;">
    Réponse rapide
  </div>

  <p style="margin:0;color:#1E293B;
            font-size:clamp(1rem,1.5vw,1.1rem);
            line-height:1.7;font-weight:500;">
    Votre réponse de 40 à 80 mots ici. Format factuel-neutre,
    contient les entités clés, les chiffres en euros, et les
    marqueurs France/Toulouse/2026.
  </p>

</div>
```

Le badge orange "Réponse rapide" en haut signale visuellement au lecteur (et structurellement aux LLM via le contexte sémantique de la classe et du contenu) que ce bloc est la synthèse extractible.

5. Les 5 règles d'or pour adapter ce système

Si vous voulez créer votre propre catalogue ou adapter le mien à votre charte graphique, voici les règles que j'ai apprises et qui valent leur pesant d'or :

1. **Une couleur primaire et une seule** : chez moi, c'est l'orange `#FF7849`. Toute la cohérence visuelle vient de ce choix répété systématiquement.
2. **Un fond sombre pour les blocs de focus** (hero, TL;DR, social, CTA) : `#0A1A2F` chez moi. Tranche avec le corps de page blanc et accroche le regard.
3. **Une typographie unique** : Lato chez moi. Pas de mélange Lato + Inter + Roboto qui dilue l'identité.

4. **Les CTA toujours en orange vif** : jamais en bleu, jamais en vert. Le lecteur doit identifier en moins d'une seconde où cliquer.
5. **Les chiffres clés toujours en orange** : jamais en vert (qui évoque le marketing positif), jamais en violet. L'orange est associé à la donnée signature de la marque.

> **Conseil pratique** : ne refaites pas mon catalogue depuis zéro. Le code complet de mon Skill (chapitre 7) inclut tous les modules, vous n'aurez qu'à les adapter à vos couleurs et votre logo.

CHAPITRE 6

Schema JSON-LD : le langage que parlent les LLM

Le Schema JSON-LD est l'élément technique le plus sous-utilisé du web SEO français. C'est aussi l'un des plus puissants pour le GEO en 2026. Voici comment l'implémenter proprement.

1. Pourquoi le Schema est devenu non-négociable

Le Schema (anciennement Schema.org) est un vocabulaire standardisé qui décrit le contenu de votre page dans un format que les machines comprennent sans ambiguïté. Quand un LLM ou un moteur de recherche lit votre page, il s'appuie sur le Schema pour comprendre :

- De quel type de contenu il s'agit (Article, FAQ, Recette, Service...)
- Qui en est l'auteur (votre nom, votre métier, votre site)
- Quand il a été publié et mis à jour
- Quels sont les éléments structurés (questions, étapes, prix, durée)

Sans Schema, le moteur doit deviner tout ça à partir du HTML, et il se trompe régulièrement. Avec Schema, vous lui donnez la carte d'identité de votre page directement.

2. Les 4 Schemas obligatoires en 2026

Schema Article (toujours)

Le Schema Article identifie le contenu comme un article de blog ou une page éditoriale. Il contient l'auteur, la date de publication, l'image principale, le titre, la description.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Article",
  "headline": "Titre exact de la page",
  "description": "Description en 150-155 caractères",
  "author": {
    "@type": "Person",
    "name": "Lucas Fonseque",
    "jobTitle": "Consultant SEO & IA",
```

```
    "url": "https://lucasfonseque.fr/",
    "sameAs": [
      "https://www.linkedin.com/in/lucas-fonseque/",
      "https://www.youtube.com/@lucas-fonseque"
    ]
  },
  "publisher": {
    "@type": "Organization",
    "name": "lucasfonseque.fr",
    "url": "https://lucasfonseque.fr/"
  },
  "datePublished": "2026-04-25T18:00:00+02:00",
  "dateModified": "2026-04-25T18:00:00+02:00",
  "image": "https://lucasfonseque.fr/wp-content/uploads/...",
  "mainEntityOfPage": {
    "@type": "WebPage",
    "@id": "https://lucasfonseque.fr/[slug]/"
  },
  "keywords": "skill claude geo, GEO France, Generative Engine Optimization, 2026"
}
</script>
```

Notez le bloc `author` détaillé avec `jobTitle`, `url`, et `sameAs` (vos profils sociaux). C'est l'un des éléments les plus forts pour l'E-E-A-T moderne.

Schema FAQPage (si vous avez une FAQ)

Le Schema FAQPage est synchronisé avec votre FAQ visible. Chaque question / réponse devient un bloc structuré.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "FAQPage",
  "mainEntity": [
    {
      "@type": "Question",
      "name": "Quelle est la différence entre un Skill Claude et un Custom GPT ?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Un Skill Claude et un Custom GPT répondent au même besoin..."
      }
    },
    {
      "@type": "Question",
      "name": "Faut-il savoir coder pour créer un Skill Claude ?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Pas obligatoirement..."
      }
    }
```

```
    // ... 8 autres questions
  ]
}
</script>
```

Important : le texte des réponses dans le Schema doit correspondre exactement (à la mise en forme près) au texte visible dans votre FAQ HTML. Google et les LLM pénalisent les Schemas qui mentionnent du contenu absent du HTML rendu.

Schema BreadcrumbList (toujours)

Le breadcrumb décrit le fil d'Ariane de la page. Très utile pour le SEO (rich snippets) et pour aider les LLM à comprendre la hiérarchie de votre site.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "BreadcrumbList",
  "itemListElement": [
    {
      "@type": "ListItem",
      "position": 1,
      "name": "Accueil",
      "item": "https://lucasfonseque.fr/"
    },
    {
      "@type": "ListItem",
      "position": 2,
      "name": "Claude IA",
      "item": "https://lucasfonseque.fr/claude-ia/"
    },
    {
      "@type": "ListItem",
      "position": 3,
      "name": "Skill Claude GEO 2026",
      "item": "https://lucasfonseque.fr/claude-ia/skill-geo-2026/"
    }
  ]
}
</script>
```

Schema HowTo (pour les tutoriels)

Si votre article est un tuto avec des étapes, ajoutez un Schema HowTo qui décrit chaque étape. Il alimente directement les rich snippets type "comment faire X" dans Google et les réponses structurées des LLM.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "HowTo",
  "name": "Comment créer un Skill Claude pour le GEO",
  "totalTime": "PT2H",
  "step": [
    {
      "@type": "HowToStep",
      "position": 1,
      "name": "Créer le dossier du Skill",
      "text": "Créez un dossier nommé selon votre cas d'usage..."
    },
    {
      "@type": "HowToStep",
      "position": 2,
      "name": "Rédiger le SKILL.md",
      "text": "Le fichier SKILL.md est lu par Claude au démarrage..."
    }
    // ... autres étapes
  ]
}
</script>
```

3. Comment placer les Schemas dans votre page

Les blocs `<script type="application/ld+json">` se placent à la fin du HTML de votre page, juste avant la fermeture du conteneur principal. WordPress les accepte sans problème, à condition que votre scan sécurité ne les bloque pas (voir chapitre 8 sur les bugs WordPress).

Si vous utilisez Rank Math comme plugin SEO, vous pouvez aussi laisser Rank Math gérer une partie des Schemas (Article, Person) automatiquement, et vous occuper manuellement du FAQPage, BreadcrumbList et HowTo qui demandent une logique custom. C'est ce que je fais sur lucasfonseque.fr.

4. Vérifier vos Schemas avec les outils Google

Avant de publier, validez systématiquement vos Schemas avec :

- **Google Rich Results Test** : search.google.com/test/rich-results. Vérifie que vos Schemas sont valides et éligibles aux rich snippets.
- **Schema Markup Validator** : validator.schema.org. Validation plus stricte du vocabulaire Schema.
- **Bing Markup Validator** dans Bing Webmaster Tools. Important car ChatGPT s'appuie sur Bing pour son retrieval.

Un Schema invalide ne sera pas pris en compte par Google et risque même de pénaliser votre page. Validez avant chaque publication, c'est non-négociable.

5. Les pièges Schema à éviter

Schema mentionnant du contenu absent du HTML

C'est le piège classique : vous générez un Schema FAQPage avec des questions / réponses, mais en réalité votre HTML visible ne contient pas toutes ces questions. Google détecte la désynchronisation et désactive le rich snippet, voire pénalise la page entière.

Multiplier les Schemas redondants

Empiler 8 Schemas différents sur une même page (Article + FAQPage + HowTo + Service + Product + Course + Event + LocalBusiness) n'est pas forcément un avantage. Au contraire, ça brouille le signal. Restez sur 3 à 5 Schemas pertinents pour le type de page.

Schema imbriqué incorrect

Le Schema permet d'imbriquer des objets (un Article qui contient un author Person qui appartient à une Organization). Mais l'imbrication suit des règles strictes du vocabulaire Schema.org. Validez toujours avec les outils mentionnés plus haut.

Caractères spéciaux non échappés

Les apostrophes courbes, les guillemets, les caractères accentués doivent être correctement encodés dans le JSON. Un caractère mal échappé invalide tout le Schema. Mon conseil : générez le Schema en Python avec `json.dumps(..., ensure_ascii=False)` et écrivez-le dans le HTML sans le retoucher manuellement.

> **À retenir** : le Schema JSON-LD est un multiplicateur silencieux. Il ne change pas l'apparence de votre page mais double ou triple vos chances d'être cité dans les LLM et de gagner des rich snippets dans Google. Sur les pages stratégiques, c'est non-négociable.

CHAPITRE 7

Le Skill Claude : automatiser la production

On rentre dans le concret. Voici comment construire votre propre Skill Claude pour publier des pages SEO+GEO en quelques minutes. Je vous livre l'architecture exacte que j'utilise au quotidien.

1. Qu'est-ce qu'un Skill Claude exactement

Un Skill Claude est un dossier structuré qui contient un fichier `SKILL.md` (la documentation principale lue par Claude au démarrage) et des fichiers complémentaires : scripts Python, templates Markdown, références de configuration.

Anthropic a introduit cette fonctionnalité en 2025 pour permettre à Claude de charger automatiquement un contexte expert sur une tâche complexe et répétée. Quand vous démarrez une conversation avec Claude, le moteur charge automatiquement les Skills disponibles dans votre environnement. Si vous demandez "crée-moi un article SEO sur le sujet X", Claude reconnaît la tâche, ouvre le SKILL.md correspondant, et applique les règles que vous y avez codifiées.

C'est radicalement différent d'un prompt système : les règles sont structurées dans des fichiers versionnés, partagés entre projets, et chargés à la demande sans saturer la fenêtre de contexte.

2. Pourquoi un Skill change la donne pour votre activité

Si vous produisez régulièrement du contenu SEO, un Skill bien construit transforme votre activité de trois façons.

Vitesse de production

Sur lucasfonseque.fr, je suis passé de **4 heures par article** (rédaction, FAQ, Schemas, mise en forme, publication) à **1h30 effectives**, dont 30 secondes de publication automatisée. C'est une économie de 60% du temps de production.

Sur 50 articles par an, ça représente environ 125 heures récupérées. À 80 € HT de l'heure facturée à un client, c'est 10 000 € de capacité supplémentaire annuelle. Ou simplement 125 heures pour faire autre chose, comme prospecter ou former.

Cohérence éditoriale

Avec un Skill, 100% des nouveaux articles respectent les mêmes règles structurelles : FAQ 10 questions × 200 mots, Schemas JSON-LD valides, marqueurs France/Toulouse/2026, ton "je" dans les paragraphes narratifs et factuel-neutre dans les blocs extractibles. Plus de pages "moins bien optimisées" parce que c'était un mauvais jour ou un nouveau rédacteur.

Évolutivité du système

Quand les LLM évoluent (et ils évoluent vite : GPT-5.4 a changé les patterns de citation en avril 2026, Claude 5 sortira en 2026), vous mettez à jour votre SKILL.md à un seul endroit, et toutes vos prochaines productions intègrent immédiatement les nouvelles règles. Sans Skill, vous devriez réécrire vos templates Word, mettre à jour vos process, briefer vos rédacteurs, etc.

-60%	100%	2h
TEMPS PAR ARTICLE	COHÉRENCE FAQ + SCHEMAS	POUR INSTALLER LE SYSTÈME

3. Architecture du Skill

Voici l'arborescence type d'un Skill GEO. C'est exactement l'architecture que j'utilise sur mon environnement :

```
geo-claude-france/
├── SKILL.md                          # Doc principale lue par Claude
├── scripts/
│   ├── cocon_helpers.py              # API publication WordPress
│   ├── template_new_page.py          # Template prêt à dupliquer
│   ├── verify_page.py                # Vérification post-publication
│   └── security_scan_patch.py        # Scan sécurité v2 (autorise JSON-LD)
└── references/
    ├── modules_catalog.py            # 18 modules HTML prêts à l'emploi
    ├── module_16_geo_additions.py    # Modules 16, 17, 18 (GEO 2026)
    ├── schema_markup.py              # Générateurs Schema JSON-LD
```

```
    ├── editorial_rules.md          # Règles éditoriales détaillées
    ├── geo_optimization.md         # Guide GEO 2026 complet
    └── wordpress_config.py         # Config WordPress (cocons, IDs)
```

4. Le fichier SKILL.md : la pièce maîtresse

Le `SKILL.md` est lu par Claude au démarrage de chaque conversation. Il doit contenir, dans cet ordre :

1. **Un en-tête YAML** avec le nom du Skill et sa description (Claude utilise la description pour décider quand activer le Skill).
2. **Les règles non-négociables** : ce qui ne doit jamais être violé, avec des exemples de bugs si on viole la règle.
3. **Le pattern minimal** en code Python copiable, qui montre comment composer une page de A à Z.
4. **Les règles éditoriales** : ton, vouvoiement, mots interdits, longueur des excerpts.
5. **Le workflow en étapes** : comprendre la demande, vérifier les doublons, composer, publier, valider.
6. **Les règles GEO spécifiques** : réponse directe, marqueurs géo-temporels, Schemas obligatoires.
7. **Les références** : liens vers les fichiers détaillés (modules_catalog, schema_markup, etc.).

5. Le code des helpers Python

Le cœur du Skill, ce sont les helpers Python qui automatisent les tâches répétitives. Voici les fonctions principales que vous devez avoir.

publish_with_divi_wrapper : l'API principale

Cette fonction encapsule tout le pipeline de publication. Elle wrappe votre HTML dans le bon conteneur Divi (avec les bonnes options qui évitent les bugs WordPress du chapitre 8), active Divi Builder, ferme commentaires et pings, et publie.

```
def publish_with_divi_wrapper(page_id, inner_html, **fields):
    """
    Publie/met à jour une page avec le wrapper Divi correct.

    Garanties :
    - Divi Builder forcé ON (meta _et_pb_use_builder=on)
    - HTML wrappé dans [et_pb_code] (pas [et_pb_text]) -> pas de wpautop
    - Largeur 90% / max 1200px
    - Comments + pings closed
```

```
    """
    wrapped = wrap_in_divi_code(inner_html)
    base_meta = fields.pop("meta", {}) or {}
    base_meta["_et_pb_use_builder"] = "on"
    fields["meta"] = base_meta
    fields.setdefault("comment_status", "closed")
    fields.setdefault("ping_status", "closed")
    return update_page(page_id, content=wrapped, **fields)


def wrap_in_divi_code(inner_html, width="90%", max_width="1200px"):
    """
    Wrappe le HTML modulaire dans une section/row/column/code Divi.

    Le module [et_pb_code] préserve le HTML brut (PAS [et_pb_text]
    qui applique wpautop !). width="90%" + max_width="1200px"
    -> contenu centré, pas de full-width écran.
    """
    return (
        f'[et_pb_section fb_built="1" admin_label="section" '
        f'_builder_version="4.27.5" _module_preset="default"]'
        f'[et_pb_row admin_label="row" _builder_version="4.27.5" '
        f'_module_preset="default" width="{width}" '
        f'max_width="{max_width}"]'
        f'[et_pb_column type="4_4" _builder_version="4.27.5" '
        f'_module_preset="default"]'
        f'[et_pb_code _builder_version="4.27.5" '
        f'_module_preset="default"]'
        + inner_html +
        '[/et_pb_code][/et_pb_column][/et_pb_row][/et_pb_section]'
    )
```

build_faq_html : FAQ avec validators automatiques

Cette fonction génère le HTML de la FAQ et applique les validators : 10 questions minimum, 200 mots minimum par réponse, découpage automatique en 3 paragraphes équilibrés pour la lisibilité.

```
def build_faq_html(faq_list, title="Questions fréquentes"):
    """
    Génère le bloc FAQ HTML avec :
    -
/
 pour l'accordéon natif
    - Chaque réponse découpée en 3 paragraphes équilibrés
    - Validator strict : 10 questions × 200 mots minimum

    Format faq_list :
        {"question": "...?", "paragraphs": ["P1", "P2", "P3"]}
        OU
        {"question": "...?", "answer": "réponse complète"}
    """
```

```
    if len(faq_list) < 10:
        raise ValueError(f"FAQ doit avoir 10 questions minimum (reçu:
{len(faq_list)})")

    items = []
    for i, qa in enumerate(faq_list, 1):
        if "paragraphs" in qa:
            paragraphs = qa["paragraphs"]
        else:
            paragraphs = split_balanced_paragraphs(qa["answer"], n_target=3)

        total_words = sum(len(p.split()) for p in paragraphs)
        if total_words < 200:
            raise ValueError(
                f"FAQ Q{i} '{qa['question'][:50]}' "
                f"fait {total_words} mots (200 min)"
            )

        # Construction des paragraphes avec spacing
        paragraphs_html = ""
        for j, p in enumerate(paragraphs):
            margin = "0 0 1rem" if j < len(paragraphs)-1 else "0"
            paragraphs_html += f'
{p}

'


        items.append(build_details_block(qa["question"], paragraphs_html))

    return wrap_in_faq_container(title, "".join(items))
```

split_balanced_paragraphs : découpage sémantique

Le découpage des longues réponses FAQ est crucial pour la lisibilité mobile. Cette fonction cherche les transitions sémantiques fortes (Premièrement, En revanche, Mon conseil, Pour le GEO...) et coupe en priorité à ces endroits, tout en gardant l'équilibre en taille.

```
def split_balanced_paragraphs(text, n_target=3):
    """
    Découpe une réponse FAQ en N paragraphes équilibrés.
    Préfère couper aux transitions sémantiques fortes.
    """
    sentences = re.split(r'(?<=[.!?])\\s+', text.strip())
    if len(sentences) <= n_target:
        return [text]

    total_words = sum(len(s.split()) for s in sentences)
    target = total_words / n_target

    breakers = (
        'Premièrement', 'Deuxièmement', 'Troisièmement',
        'Première erreur', 'Deuxième erreur', 'Troisième erreur',
```

```
        'Premier marqueur', 'Deuxième marqueur',
        "À l'inverse", 'En revanche', 'En pratique',
        'Mon conseil', 'Mon choix',
        'Pour accélérer', 'Pour le GEO', 'Pour comparer',
        'Si vous', 'Surveillez aussi',
    )

    paragraphs, current, current_words = [], [], 0
    for s in sentences:
        s_words = len(s.split())
        if len(paragraphs) == n_target - 1:
            current.append(s)
            current_words += s_words
            continue
        is_breaker = any(s.startswith(b) for b in breakers)
        zone_ok = current_words >= target * 0.7
        zone_must_cut = current_words >= target * 1.1
        if zone_ok and is_breaker and current:
            paragraphs.append(' '.join(current))
            current = [s]
            current_words = s_words
            continue
        if zone_must_cut and current:
            paragraphs.append(' '.join(current))
            current = [s]
            current_words = s_words
            continue
        current.append(s)
        current_words += s_words

    if current:
        paragraphs.append(' '.join(current))
    return paragraphs
```

img_to_background_div : la solution anti-bug cartes

L'un des bugs les plus pénibles de WordPress (détaillé au chapitre 8) est que les `<img>` à l'intérieur d'un `<a>` sont attaquées par le couple lazy-load + `<noscript>`, ce qui casse les liens parents et rend les cartes "vides" visuellement.

La solution : remplacer les `<img>` par des `<div>` avec `background-image`. Sémantiquement équivalent (avec l'attribut `role="img"` et un `aria-label`), mais invisible au lazy-load.

```
def img_to_background_div(img_url, alt_text, aspect="16/9"):
    """
    Génère un
 au lieu d'un .
    Évite le couple lazy-load + wpautop qui casse les  parents.
    """
    return (
        f'
```

```
'
    )
```

6. Le fichier modules_catalog.py

Ce fichier contient les 18 modules HTML pré-écrits, exportés comme des constantes Python. Vous les importez et les concaténez. Voici un extrait pour le module 08 (réseaux sociaux), tel qu'il apparaît dans mon catalogue :

```
MOD_08_SOCIAL = '''
<div class="lf-mod-08-social"
     style="background:#0F172A;border-radius:1.25rem;
            padding:clamp(2rem,4vw,3rem) clamp(1rem,3vw,2rem);
            color:white;text-align:center;margin:2rem 0;">
  <h3 style="margin:0 0 .75rem;color:white;
             font-size:clamp(1.25rem,2.5vw,1.75rem);
             font-weight:800;">
    Retrouvez-moi sur les réseaux
  </h3>
  <p style="color:#94A3B8;margin:0 0 2rem;
            font-size:1rem;line-height:1.6;">
    Je partage mes expérimentations SEO et IA au quotidien.
  </p>
  <div style="display:flex;gap:1rem;
              justify-content:center;flex-wrap:wrap;">
    <!-- 4 boutons réseaux ici (YouTube, TikTok, Instagram, LinkedIn) -->
  </div>
</div>
'''
```

7. Le fichier schema_markup.py

Ce fichier contient les générateurs de Schema JSON-LD. Voici la fonction principale pour un article standard :

```
def build_standard_article_schemas(
    title, description, slug, featured_image_url,
    date_published, faq_list, breadcrumb_items, keywords
):
    """
    Génère les 3 Schemas standards d'un article :
    Article + FAQPage + BreadcrumbList.
```

```
    Retourne le HTML complet avec les 3 blocs <script>.
    """
    article = build_article_schema(
        title, description, slug, featured_image_url,
        date_published, keywords
    )
    faq = build_faq_schema(faq_list)
    breadcrumb = build_breadcrumb_schema(breadcrumb_items)

    return (
        f'<script type="application/ld+json">{json.dumps(article,
ensure_ascii=False)}</script>'
        f'<script type="application/ld+json">{json.dumps(faq, ensure_ascii=False)}
</script>'
        f'<script type="application/ld+json">{json.dumps(breadcrumb,
ensure_ascii=False)}</script>'
    )
```

8. Mise en pratique : votre première publication

Avec ces helpers en place, voici à quoi ressemble la publication d'une page complète :

```
import sys
sys.path.insert(0, '/votre/chemin/skill/scripts')
sys.path.insert(0, '/votre/chemin/skill/references')

import cocon_helpers as ch
from schema_markup import build_standard_article_schemas
from module_16_geo_additions import build_direct_answer
from modules_catalog import MOD_08_SOCIAL

# 1. Composer chaque module
HERO = ch.minify_html('<div class="lf-mod-01-hero" ...>...</div>')
DIRECT = ch.minify_html(build_direct_answer("Réponse 40-80 mots..."))
INTRO = ch.minify_html('<div class="lf-mod-02-title-text">...</div>')
# ... autres modules

# 2. FAQ avec validators
FAQ_BLOCK = ch.build_faq_html([
    {"question": "Q1?", "answer": "200+ mots..."},
    # ... 10 questions
])

# 3. Pages sœurs avec vraies featured images
PAGES_SOEURS = ch.build_pages_soeurs_block([
    ("https://...", "Guide", "Titre", "Desc",
     ch.get_real_featured_image("slug-cible")),
    # ... 2-3 cartes
```

```
])

# 4. Schemas JSON-LD
SCHEMA = ch.minify_html(build_standard_article_schemas(
    title=TITLE, description=META_DESC, slug=SLUG,
    featured_image_url=IMAGE_URL,
    date_published="2026-04-25T18:00:00+02:00",
    faq_list=faq_list, breadcrumb_items=breadcrumbs,
    keywords=[FOCUS_KEYWORD, "France", "2026"]
))

# 5. Concaténer tout
inner = HERO + DIRECT + INTRO + ... + FAQ_BLOCK + PAGES_SOEURS + SCHEMA

# 6. Publier (1 seul appel pour tout faire)
ch.publish_with_divi_wrapper(page_id, inner, excerpt=EXCERPT)

# 7. Configurer Rank Math (object_type="post" même pour les pages)
ch.rank_math(page_id, FOCUS_KEYWORD, META_TITLE, META_DESC,
             object_type="post")
```

En 7 étapes, votre page complète est publiée avec FAQ validée, Schemas JSON-LD générés, modules HTML cohérents, métadonnées Rank Math configurées. 30 secondes une fois le contenu rédigé.

Le game-changer : ce système n'est pas qu'un gain de temps. C'est un filtre qualité. Le validator FAQ refuse de publier si une réponse fait moins de 200 mots. Le scan sécurité refuse les balises interdites. Le wrapper Divi évite les bugs visuels. Vos publications sont systématiquement aux normes, sans dépendre de votre concentration ce jour-là.

CHAPITRE 8

Les bugs WordPress qui cassent vos pages

WordPress est une excellente plateforme, mais elle a des pièges spécifiques quand on injecte du HTML modulaire avancé. Voici les 5 bugs que j'ai rencontrés en production et leurs correctifs validés. Apprendre ces leçons en lisant ce guide vous évitera plusieurs heures de debug.

1. Bug 1 : le H1 doublé

Symptôme : le titre de votre page apparaît deux fois sur le rendu front. Une fois en haut (gros titre du thème), une fois dans le hero (votre H1 modulaire).

Cause : les thèmes WordPress (notamment Divi) affichent leur propre H1 généré à partir du titre de la page WP, en plus de votre HTML personnalisé. Ce H1 du thème ne se voit que quand le builder est désactivé. Quand le builder est activé, le thème délègue l'affichage et n'ajoute pas son H1.

Correctif : forcer Divi Builder à ON via la meta `_et_pb_use_builder = "on"`. Contre-intuitif si vous pensiez que désactiver le builder permettait d'éditer en HTML pur. La logique est inverse : builder ON laisse Divi gérer l'affichage du contenu (incluant votre HTML modulaire dans un module Code), builder OFF laisse le thème prendre la main et afficher son H1 par-dessus.

```
# Activer le Divi Builder (oui, contre-intuitif)
requests.post(f"{WP_BASE}/pages/{page_id}", auth=WP_AUTH, json={
    "meta": {"_et_pb_use_builder": "on"}
})
```

2. Bug 2 : wpautop qui casse les liens cartes

Symptôme : vos cartes "pages sœurs" et "articles blog" apparaissent vides ou cassées sur le front. Le HTML que vous voyez en sortie contient des `<p></a>` orphelins.

Cause : WordPress applique `wpautop` au contenu des modules `[et_pb_text]`. Cette fonction remplace les sauts de ligne par des `<br>` et entoure les blocs de `<p>`, ce qui casse les `<a>` parents qui contiennent du HTML structuré.

Correctif : utiliser le module `[et_pb_code]` au lieu de `[et_pb_text]`. Le module Code de Divi préserve le HTML brut sans appliquer `wpautop`. Toujours wrapper votre HTML modulaire dans `[et_pb_code]`, jamais `[et_pb_text]`.

```
# MAUVAIS - applique wpautop, casse les liens
content = '[et_pb_text]' + my_html + '[/et_pb_text]'

# BON - préserve le HTML brut
content = '[et_pb_code]' + my_html + '[/et_pb_code]'
```

3. Bug 3 : lazy-load + noscript qui casse les cartes

Symptôme : même avec `[et_pb_code]`, certaines cartes pages sœurs ou blog apparaissent comme des cadres vides sur le front. Le HTML montre des `<noscript>` insérés au milieu de vos `<a>`.

Cause : les plugins de lazy-load WordPress (intégrés à Jetpack, Rocket, Smush, Optimole) transforment vos `<img>` en double balise `<img>` + `<noscript>`. Si cette transformation se fait à l'intérieur d'un `<a>`, le parseur HTML du navigateur interprète mal et ferme le `<a>` au mauvais endroit, ce qui casse la carte cliquable.

Correctif : remplacer les `<img>` dans les cartes par des `<div>` avec `background-image`. Sémantiquement équivalent (avec `role="img"` et `aria-label`), mais invisible au plugin lazy-load.

```
# MAUVAIS - sera attaqué par le lazy-load
'<a href="..."><img src="..." alt="..." />...</a>'

# BON - immune au lazy-load
'<a href="...">'
'  <div role="img" aria-label="..." '
'       style="background-image:url(...);'
'              background-size:cover;'
```

```
'              aspect-ratio:16/9"></div>'
'  ...'
'</a>'
```

4. Bug 4 : full-width au lieu de centré 90%

Symptôme : votre contenu prend toute la largeur de l'écran, sans aération. Sur grand écran, les lignes de texte deviennent illisibles (200+ caractères de large).

Cause : les sections Divi par défaut sont en `fb_built="1"` avec largeur 100%. Sans contrainte explicite, votre contenu débord du conteneur classique du thème.

Correctif : forcer la largeur sur la `[et_pb_row]` avec `width="90%"` et `max_width="1200px"`. C'est l'équivalent visuel d'un container Bootstrap centré.

```
# Wrapper Divi correct avec largeur 90% / max 1200px
'[et_pb_section fb_built="1" '
' _builder_version="4.27.5"]'
'[et_pb_row '
' width="90%" max_width="1200px" '
' _builder_version="4.27.5"]'
'[et_pb_column type="4_4" _builder_version="4.27.5"]'
'[et_pb_code _builder_version="4.27.5"]'
+ inner_html +
'[/et_pb_code][/et_pb_column][/et_pb_row][/et_pb_section]'
```

5. Bug 5 : Rank Math 403 sur object_type="page"

Symptôme : votre script de configuration Rank Math renvoie une erreur HTTP 403 "rest_forbidden" sur les pages, alors qu'il fonctionne sur les articles.

Cause : l'endpoint Rank Math `/wp-json/rankmath/v1/updateMeta` n'accepte pas `object_type="page"`. Comportement non documenté qui m'a coûté plusieurs heures de debug.

Correctif : utiliser `object_type="post"` même pour configurer une page. Rank Math gère les deux types via le même endpoint mais ne discrimine pas correctement.

```
# MAUVAIS - HTTP 403
ch.rank_math(page_id, focus_keyword, title, description,
             object_type="page")

# BON - fonctionne pour pages ET articles
ch.rank_math(page_id, focus_keyword, title, description,
             object_type="post")
```

6. Vérification post-publication automatique

Pour ne plus jamais subir ces bugs en silence, intégrez une vérification automatique en fin de pipeline. Voici le code que j'utilise après chaque publication :

```
import requests, re, time
time.sleep(3)  # laisser le cache se rafraîchir

r = requests.get(
    f"https://votresite.fr/{slug_complet}/",
    headers={"User-Agent": "Mozilla/5.0"},
    timeout=30
)
html = r.text

# Vérification 1 : un seul H1
nb_h1 = len(re.findall(r'<h1[^>]*>', html))
assert nb_h1 == 1, f"H1 dupliqué ! ({nb_h1} trouvés)"

# Vérification 2 : pas de  orphelins (= cartes cassées)
nb_orphans = len(re.findall(r'<p>\\s*</a>', html, re.DOTALL))
assert nb_orphans == 0, f"Cartes cassées par wpautop ({nb_orphans})"

# Vérification 3 : Schemas JSON-LD bien injectés
nb_schemas = html.count('application/ld+json')
assert nb_schemas >= 3, f"Schemas manquants ({nb_schemas}/3 attendus)"

print("Page saine - publication validée")
```

Si l'une des assertions échoue, vous savez immédiatement quel bug s'est invité, et vous pouvez corriger sans attendre le retour utilisateur.

7. Le scan sécurité WordPress et JSON-LD

Dernier piège : par défaut, les scans sécurité WordPress (Wordfence, iThemes, Securi) bloquent les balises `<script>` non standards. Or les Schemas JSON-LD utilisent `<script type="application/ld+json">`.

Solution : patcher votre scan sécurité pour autoriser spécifiquement le type MIME `application/ld+json`. Voici le code de mon `security_scan_v2()` :

```
def security_scan_v2(html):
    """
    Scan sécurité v2 qui autorise les blocs JSON-LD Schema.
    Bloque toujours les <script> classiques, <iframe>, <style>, etc.
    """
    # Blacklist standard
    forbidden = [
        r'<script(?!\\s+type="application/ld\\+json")[^>]*>',
        r'<iframe', r'<style', r'<head',
        r'<meta\\s+(?!name=)', r'on(click|load|error|mouseover)='
    ]

    for pattern in forbidden:
        if re.search(pattern, html, re.IGNORECASE):
            return False

    # Compte les blocs JSON-LD valides (info)
    json_ld_count = len(re.findall(
        r'<script\\s+type="application/ld\\+json">',
        html
    ))
    print(f"  Scan OK : {json_ld_count} bloc(s) JSON-LD Schema valide(s)")
    return True
```

À retenir : ces 5 bugs ont chacun coûté des heures de debug à les identifier la première fois. Les avoir documentés dans votre Skill (avec le correctif automatique dans les helpers) garantit qu'ils ne se reproduiront plus jamais. C'est l'un des bénéfices cachés les plus précieux de l'industrialisation par Skill.

CHAPITRE 9

Le pipeline complet : de la commande à la publication

On assemble tout. Voici le pipeline end-to-end que j'utilise quotidiennement pour publier une page complète. C'est le code exact, exécutable, que vous pouvez adapter à votre environnement.

1. Vue d'ensemble du pipeline

Le pipeline complet enchaîne 7 étapes :

1. Construction de la FAQ avec validators (10 Q × 200 mots)
2. Composition du HTML modulaire (16 modules + Schemas)
3. Scan sécurité v2 (autorise JSON-LD)
4. Vérification anti-doublon par slug
5. Publication via le wrapper Divi (Builder ON, et_pb_code, 90%/1200px)
6. Configuration Rank Math (focus, title, description)
7. Vérification automatique du rendu front

2. Le code complet du pipeline

Voici le code `template_new_page.py` dans son intégralité. Vous le copiez sous un nouveau nom, vous remplissez la config et le contenu, vous lancez. C'est tout.

```
"""
template_new_page.py
TEMPLATE PRÊT À DUPLIQUER pour créer une nouvelle page cocon.

Pattern validé sur la page 3123 (avril 2026), encapsule les 5
règles non-négociables :
  1. Divi Builder ON (sinon le thème dédouble le H1)
  2. Module [et_pb_code] (pas [et_pb_text]) - pas de wpautop
  3. Largeur 90% / max 1200px (sinon full-width)
  4.
 dans les cartes
  5.  au lieu de
```

```
 dans les cartes
'''
import sys, time, requests, re

sys.path.insert(0, '/votre/chemin/skill/scripts')
sys.path.insert(0, '/votre/chemin/skill/references')

import cocon_helpers as ch
from security_scan_patch import security_scan_v2
from schema_markup import build_standard_article_schemas
from module_16_geo_additions import build_direct_answer
from modules_catalog import MOD_08_SOCIAL


# ======================================================
# CONFIG DE LA PAGE - À REMPLIR
# ======================================================
SLUG = "mon-nouveau-slug-2026"
TITLE = "Mon titre H1 complet pour le hero"
PARENT_ID = 1860  # ID du cocon parent
FEATURED_MEDIA_ID = 1901  # Image par défaut

FOCUS_KEYWORD = "mon focus keyword"
META_TITLE = "Mon Meta Title (max 60 chars)"
META_DESCRIPTION = "Ma meta description sous 115 caractères."
EXCERPT = "Mon excerpt de 150-155 caractères."

DATE_PUBLISHED = "2026-04-25T18:00:00+02:00"
SLUG_FULL_PATH = f"claude-ia/{SLUG}"


# ======================================================
# CONSTRUCTION DES MODULES
# ======================================================

# Module 01 - Hero
HERO = ch.minify_html(f'''
<div class="lf-mod-01-hero" style="...">
  <div>Catégorie - 2026</div>
  <h1>{TITLE_PART_1}<br>
    <span class="gradient">{TITLE_PART_2}</span>
  </h1>
  <p>Sous-titre de la promesse</p>
  <div class="stats">...</div>
</div>
''')

# Module 16 - Réponse directe (NOUVEAU GEO)
DIRECT_ANSWER = ch.minify_html(build_direct_answer(
    "Réponse 40-80 mots avec entités clés, chiffres en euros, "
    "et marqueurs France/Toulouse/2026."
))

# Intro "je"
INTRO_JE = ch.minify_html('''
```

```
<div class="lf-mod-02-title-text">
  <h2>Pourquoi je vous parle de ce sujet</h2>
  <p>Premier paragraphe ton "je"...</p>
  <p>Deuxième paragraphe avec chiffre bénéfice...</p>
</div>
''')

# H2s (5 dimensions du fan-out GEO)
H2_DEFINITION = h2_block("Qu'est-ce que [sujet] ?", [...])
H2_FONCTIONNEMENT = h2_block("Comment ça fonctionne ?", [...])
H2_PRIX = h2_block("Combien ça coûte en France en 2026 ?", [...])
H2_CIBLE = h2_block("Qui devrait s'y intéresser ?", [...])

# Tableau comparatif (Module 18)
COMPARATIF = ch.minify_html(build_comparison_table(
    headers=["Approche", "Temps", "Coût", "Niveau", "Adapté pour"],
    rows=[
        ["Option A", "X heures", "Y EUR", "Niveau", "Profil"],
        # ... 2-3 lignes
    ],
    caption="Comparatif en France en 2026"
))

# Cas concret chiffré
CAS_CONCRET = ch.minify_html('''
<div class="lf-mod-10-callout">
  <h3>Mon cas concret</h3>
  <ul>
    <li>Métrique 1 : <strong style="color:#FF7849">+47%</strong></li>
    <li>Métrique 2 : économie de 60% du temps</li>
    <li>Métrique 3 : 0 -> 3 résultats en 8 semaines</li>
  </ul>
</div>
''')

# CTA Calendly
CTA = ch.minify_html('''
<div class="lf-mod-06-contact">
  <img src="..." alt="Photo Lucas Fonseque" />
  <h2>Vous voulez votre Skill GEO sur mesure ?</h2>
  <p>30 minutes pour caler votre stratégie.</p>
  <a href="https://calendly.com/...">Réserver mon créneau</a>
</div>
''')

# FAQ avec validators (10 Q x 200 mots)
FAQ_QUESTIONS = [
    {"question": "Q1 ?", "answer": "200+ mots..."},
    # ... 9 autres questions
]

# TL;DR (Module 17)
TLDR = ch.build_tldr_block([
    "Premier point clé",
    "Deuxième point",
```

```
    # ... 5-7 bullets
], label="TL;DR - à retenir")

# Pages sœurs avec vraies featured images (Module 04)
PAGES_SOEURS = ch.build_pages_soeurs_block([
    ("https://...", "Guide", "Titre", "Desc",
     ch.get_real_featured_image("slug-cible", post_type="pages")),
    # ... 2-3 cartes
], h2_title="Pour aller plus loin sur [sujet]")

# Trustindex (séparateur)
TRUSTINDEX = '''<div class="lf-mod-15-trustindex">
  <h2>Ce que disent mes clients</h2>
  [trustindex no-registration=google]
</div>'''

# Articles blog avec vraies images (Module 05)
ARTICLES = ch.build_articles_blog_block([
    ("https://...", "Titre", "Desc",
     ch.get_real_featured_image("slug-article"), "Catégorie"),
    # ... 3 articles
], h2_title="Mes derniers articles")

# Réseaux sociaux (Module 08 - validé)
SOCIAL = ch.minify_html(MOD_08_SOCIAL)


# ============================================================
# PIPELINE DE PUBLICATION
# ============================================================
def main():
    print(f"Publication : {TITLE}")

    # 1. Construction FAQ avec validators
    try:
        FAQ_BLOCK = ch.build_faq_html(FAQ_QUESTIONS)
        print(f"  FAQ valide ({len(FAQ_QUESTIONS)} questions)")
    except ValueError as e:
        print(f"  FAQ invalide : {e}")
        return

    # 2. Schemas JSON-LD
    faq_for_schema = [
        {"question": q["question"],
         "answer": q.get("answer") or " ".join(q.get("paragraphs", []))}
        for q in FAQ_QUESTIONS
    ]
    breadcrumbs = [
        {"name": "Accueil", "url": "https://lucasfonseque.fr/"},
        {"name": "Catégorie",
         "url": "https://lucasfonseque.fr/[parent]/"},
        {"name": TITLE,
         "url": f"https://lucasfonseque.fr/{SLUG_FULL_PATH}/"},
    ]
    SCHEMAS = ch.minify_html(build_standard_article_schemas(
```

```
        title=TITLE, description=META_DESCRIPTION,
        slug=SLUG_FULL_PATH,
        featured_image_url=ch.FEATURED_IMAGE_COCON_URL,
        date_published=DATE_PUBLISHED,
        faq_list=faq_for_schema,
        breadcrumb_items=breadcrumbs,
        keywords=[FOCUS_KEYWORD, "France", "2026"]
    ))

    # 3. Composition HTML
    INNER_HTML = (
        HERO + DIRECT_ANSWER + INTRO_JE
        + H2_DEFINITION + H2_FONCTIONNEMENT + H2_PRIX + H2_CIBLE
        + COMPARATIF + CAS_CONCRET + CTA
        + FAQ_BLOCK + TLDR
        + PAGES_SOEURS + TRUSTINDEX + ARTICLES + SOCIAL
        + SCHEMAS
    )

    # 4. Scan sécurité
    if not security_scan_v2(INNER_HTML):
        print("  Scan KO - balise interdite")
        return

    # 5. Anti-doublon
    existing = ch.search_pages_by_slug(SLUG)
    if existing:
        page_id = existing[0]["id"]
        print(f"  Page existante (ID {page_id}) - mise à jour")
    else:
        page_id = ch.create_draft_page(
            slug=SLUG, title=TITLE, content="",
            parent=PARENT_ID, featured_media=FEATURED_MEDIA_ID
        )
        print(f"  Brouillon créé : ID {page_id}")

    # 6. Publication via wrapper Divi
    ch.publish_with_divi_wrapper(
        page_id, INNER_HTML,
        excerpt=EXCERPT, status="publish"
    )

    # 7. Rank Math (object_type="post" même pour les pages)
    ch.rank_math(
        object_id=page_id,
        focus_keyword=FOCUS_KEYWORD,
        title=META_TITLE, description=META_DESCRIPTION,
        object_type="post"
    )

    # 8. Vérification automatique
    time.sleep(3)
    page_url = f"https://lucasfonseque.fr/{SLUG_FULL_PATH}/"
    r = requests.get(page_url,
                     headers={"User-Agent": "Mozilla/5.0"},
```

```
                        timeout=30)
    html = r.text

    nb_h1 = len(re.findall(r'<h1[^>]*>', html))
    nb_orphans = len(re.findall(r'<p>\\s*</a>', html,
                                re.DOTALL))

    print(f"  H1 : {nb_h1} (attendu : 1)")
    print(f"  Orphelins : {nb_orphans} (attendu : 0)")

    if nb_h1 == 1 and nb_orphans == 0:
        print("  Page saine !")
    else:
        print("  Bugs détectés - vérifier le wrapper Divi")

    print(f"\\nPublication terminée : {page_url}")


if __name__ == "__main__":
    main()
```

3. Adapter le pipeline à votre stack

Mon pipeline est calibré pour WordPress + Divi + Rank Math, qui est un combo très répandu en France. Si votre stack est différente, voici comment adapter.

WordPress + Gutenberg (sans Divi)

Si vous utilisez Gutenberg natif sans builder, vous pouvez publier votre HTML directement sans wrapper Divi. Le bloc HTML custom de Gutenberg accepte le HTML brut. Vos modules s'y insèrent tels quels.

Conséquence : vous n'avez pas le bug du H1 doublé (le thème Gutenberg gère ça correctement) ni le bug wpautop. Vous pouvez simplifier `publish_with_divi_wrapper` en `publish_with_gutenberg` qui ajoute juste les balises Gutenberg HTML.

WordPress + Elementor

Elementor a son propre système de templates. Le plus simple est d'utiliser un widget HTML ou Code custom et d'y injecter votre HTML modulaire. Pas de wrapper spécifique nécessaire.

Headless / Next.js / Astro

Si vous travaillez en headless avec Next.js, Astro, ou tout framework JS moderne, vos modules HTML deviennent des composants React/Astro/Vue. Le principe reste identique : catalogue réutilisable, validators FAQ, génération Schemas. Mais l'implémentation passe par des composants typés au lieu de chaînes HTML.

Webflow / Framer / autres CMS no-code

Plus délicat car ces plateformes restreignent l'injection HTML. Vous pouvez insérer du HTML custom dans un bloc Embed mais avec des limites de taille. Le mieux est d'embarquer le HTML modulaire en plusieurs blocs Embed séparés (un par module). Le Skill Claude vous génère le code, vous le copiez/collez dans les blocs.

4. Mesurer le ROI du pipeline

Pour estimer ce que ce système vous fait gagner, comptez votre temps actuel par article (rédaction + mise en forme + FAQ + Schemas + publication) et comparez avec ce que vous obtiendrez après mise en place.

ÉTAPE	TEMPS AVANT	TEMPS APRÈS SKILL
Rédaction du contenu	2h	1h (l'IA aide)
Mise en forme HTML	45 min	5 min (modules prêts)
FAQ structurée	30 min	15 min (template)
Schemas JSON-LD	20 min	0 (auto-généré)
Publication WordPress	15 min	30 secondes
Vérification finale	10 min	auto (assertions)
Total	**4h**	**1h30**

À 80 € HT de l'heure facturée, l'économie est de 200 € par article. Sur 50 articles annuels, c'est 10 000 € HT de capacité libérée. Le système se rentabilise dès la deuxième semaine d'utilisation.

> **L'effet composé** : au-delà du gain direct, vous publiez plus, donc vous générez plus de trafic, donc plus de leads, donc plus de chiffre d'affaires. Mes clients qui ont déployé ce système voient en moyenne +40% d'articles publiés par mois à effort constant. C'est ça, le vrai effet de levier.

CHAPITRE 10

Plan d'action 30 jours : de zéro à un système qui produit tout seul

On termine sur du 100% opérationnel. Voici le plan d'action sur 30 jours pour passer de votre situation actuelle (production manuelle, pas de GEO, pas de Skill) à un système industrialisé qui produit des pages SEO+GEO en moins de 2 heures de A à Z. Vous pouvez suivre ce plan en parallèle de votre activité normale.

1. Semaine 1 : audit de l'existant et préparation

Jour 1-2 : audit GEO de votre site actuel

Checklist audit GEO

- ☐ Vérifier votre robots.txt — autorise-t-il OAI-SearchBot, ChatGPT-User, PerplexityBot, Claude-SearchBot, Claude-User, Google-Extended ?
- ☐ Vérifier Cloudflare AI Bot Blocking — désactivé ?
- ☐ Vérifier que votre contenu est bien rendu côté serveur (pas JavaScript-only)
- ☐ Tester 10 requêtes prioritaires manuellement dans ChatGPT, Perplexity, Gemini, Claude — votre marque apparaît-elle ?
- ☐ Compter les pages qui ont déjà un Schema Article + FAQPage + BreadcrumbList
- ☐ Identifier vos 5 pages les plus stratégiques (top trafic ou top conversion)

Cet audit prend 4 à 6 heures bien faites. Notez tout dans un document de référence : vous y reviendrez à la fin du mois pour mesurer la progression.

Jour 3-4 : choix de votre stack technique

Posez-vous trois questions :

1. **Sur quel CMS publiez-vous ?** WordPress reste majoritaire en France. Si c'est votre cas, vous pouvez reprendre 90% de mon Skill tel quel.

2. **Quel builder utilisez-vous ?** Divi est répandu mais a les bugs documentés au chapitre 8. Si vous pouvez migrer vers GeneratePress ou Kadence, c'est un investissement qui paie sur la performance et la simplicité.
3. **Quel plugin SEO ?** Rank Math reste mon choix en 2026 pour la qualité de l'API REST et la gestion des Schemas. Yoast est solide aussi mais l'API est moins ouverte.

Jour 5-7 : création du dossier Skill

Créez l'arborescence du Skill sur votre poste de travail. Si vous utilisez Claude (recommandé), placez le dossier dans le chemin lu automatiquement par les Skills. Si vous utilisez un autre LLM, le principe reste le même mais vous devrez charger manuellement les fichiers en début de conversation.

```
mon-skill-seo-geo/
├── SKILL.md
├── scripts/
│   ├── cocon_helpers.py
│   ├── template_new_page.py
│   ├── verify_page.py
│   └── security_scan_patch.py
└── references/
    ├── modules_catalog.py
    ├── module_16_geo_additions.py
    ├── schema_markup.py
    └── editorial_rules.md
```

2. Semaine 2 : implémentation des fondations

Jour 8-10 : codage des helpers principaux

Commencez par les 4 helpers critiques fournis au chapitre 7 :

Helpers à coder en priorité

- ☐ `publish_with_divi_wrapper` (ou son équivalent pour votre stack)
- ☐ `build_faq_html` avec validators 10 Q × 200 mots
- ☐ `split_balanced_paragraphs` pour le découpage FAQ
- ☐ `img_to_background_div` pour les cartes
- ☐ `build_pages_soeurs_block` et `build_articles_blog_block`
- ☐ `build_tldr_block` avec couleurs validées
- ☐ `build_standard_article_schemas` pour les 3 Schemas JSON-LD

Jour 11-13 : adaptation du catalogue de modules

Adaptez les 18 modules HTML à votre charte graphique. Concrètement :

- Remplacez ma palette (#0A1A2F, #FF7849, #FFF6EF) par la vôtre. Une couleur primaire forte (l'équivalent de mon orange) est indispensable pour la cohérence visuelle.
- Adaptez la typographie (Lato chez moi). Choisissez une seule famille typo pour tout le site. Pas de mélange Inter + Roboto + Lato.
- Gardez la structure des modules (hero, réponse directe, timeline, comparatif, etc.). Ne réinventez pas la roue.

Jour 14 : test sur une première page

Testez le pipeline complet sur une page de test (URL non publiée publiquement). Mesurez chaque étape. Identifiez les bugs ou les frictions. Itérez jusqu'à ce que la publication soit propre du premier coup.

3. Semaine 3 : production et migration

Jour 15-17 : migration de vos 5 pages stratégiques

Reprenez les 5 pages stratégiques identifiées en semaine 1 et migrez-les vers le nouveau format. C'est l'étape qui apporte le plus de ROI rapidement : ces pages génèrent déjà du trafic, elles vont en générer beaucoup plus une fois GEO-optimisées.

Ne migrez pas tout d'un coup. Une page par jour, avec un test front complet et une vérification post-publication automatique. Si une page casse en migration, vous voulez pouvoir corriger avant d'enchaîner.

Jour 18-20 : production de 3 nouvelles pages cocon

Maintenant que le système tourne, produisez 3 nouvelles pages en utilisant 100% le Skill. Chronométrez. Vous devriez être autour de 2 heures par page maximum, contenu compris.

Jour 21 : audit Schema et tests outils Google

Validez tous vos Schemas JSON-LD avec :

- Google Rich Results Test (search.google.com/test/rich-results)
- Schema Markup Validator (validator.schema.org)
- Bing Markup Validator dans Bing Webmaster Tools

Corrigez les éventuels warnings. Un Schema valide est un Schema cité. Un Schema invalide est un Schema ignoré.

4. Semaine 4 : mesure et optimisation

Jour 22-24 : mise en place du tracking GEO

Activez les outils de mesure GEO suivants :

Tracking GEO à activer

- ☐ Google Search Console — segment trafic depuis chatgpt.com, perplexity.ai, gemini.google.com
- ☐ Bing Webmaster Tools — impressions Bing (qui alimente ChatGPT)
- ☐ Test mensuel manuel — 10 requêtes prioritaires dans les 5 LLM cibles, scoring binaire mention/pas mention
- ☐ Optionnel : abonnement Meteoria (75 € à 300 €/mois) pour tracking automatisé
- ☐ Configurer un dashboard Looker Studio combinant GSC + Bing + tracking manuel

Jour 25-27 : optimisation du Skill via les retours

Reprenez votre Skill et identifiez ce qui a posé problème pendant les 4 semaines. Les questions à vous poser :

- Quelles règles ai-je dû ajouter manuellement à chaque publication ? À ajouter au SKILL.md.
- Quels modules HTML m'ont manqué ? À ajouter au catalogue.

- Quels bugs ont émergé ? À documenter avec correctifs.
- Quelles métadonnées doivent être systématisées ? À ajouter au pipeline.

Jour 28-30 : bilan et plan trimestriel

Reprenez votre audit du jour 1 et mesurez la progression sur les indicateurs clés :

- Nombre de pages avec Schemas complets (cible : 100% des pages stratégiques)
- Temps moyen de production par page (cible : -60% par rapport à avant)
- Nombre de citations dans les 5 LLM (cible : au moins 1 mention sur les requêtes commerciales prioritaires)
- Trafic référent depuis les domaines IA (cible : 10+ sessions par mois)
- Trafic organique sur les pages migrées (cible : +20% à 3 mois)

Définissez ensuite un plan trimestriel sur 90 jours : nombre de pages à produire, cocons à compléter, optimisations Skill, tests A/B éventuels.

5. Erreurs à éviter pendant ces 30 jours

Vouloir tout migrer d'un coup

Si vous avez 200 pages sur votre site, ne tentez pas de toutes les migrer en 30 jours. Concentrez-vous sur les 5 à 10 pages stratégiques (top trafic ou top conversion). Le reste se fera progressivement au fil des mises à jour normales.

Trop personnaliser le système trop tôt

Beaucoup tentent d'ajouter 5 modules HTML maison avant même d'avoir publié leur première page. Résultat : le système devient complexe à maintenir, et la première publication tarde. Commencez avec 8 à 10 modules essentiels, vous ajouterez le reste plus tard quand le besoin sera précis.

Négliger les validators

La tentation est de désactiver le validator FAQ "juste pour cette page". Ne le faites pas. Les validators sont votre filet de sécurité contre les régressions de qualité. Une page sans FAQ 10 questions × 200 mots performera moins en GEO. Point.

Oublier la validation visuelle finale

Le pipeline automatique garantit la cohérence technique mais ne valide pas la qualité visuelle. Faites un test mobile + desktop sur chaque nouvelle publication. C'est 2 minutes qui vous évitent de découvrir un mois plus tard qu'une carte n'apparaît pas correctement.

6. Et après les 30 jours

Une fois le système en place et rodé, voici la cadence que je recommande pour la suite :

- **Hebdo** : 1 à 3 nouvelles publications selon votre rythme éditorial. Avec le Skill, c'est désormais tenable sans effort démesuré.
- **Mensuel** : tracking manuel des 10 requêtes prioritaires. Bilan rapide. Ajout d'1 ou 2 améliorations au Skill.
- **Trimestriel** : bilan complet des KPIs. Audit Schema. Identification des pages à mettre à jour. Réflexion stratégique sur les nouveaux cocons à créer.
- **Annuel** : refonte du Skill pour intégrer les évolutions des LLM. En 2026, attendez-vous à l'arrivée de nouveaux moteurs et à des changements dans les patterns de citation.

7. Pour aller plus loin

Cet ebook est un point de départ exhaustif, mais le SEO et le GEO évoluent vite. Voici les ressources que je consulte au quotidien pour rester à jour :

- **Mon site lucasfonseque.fr** : j'y publie mes retours d'expérience, mes tutos, mes analyses des évolutions LLM.
- **Search Engine Land** et **Search Engine Journal** : la veille SEO internationale de référence.
- **AirOps** et **Profound** : pour les données quantitatives sur les patterns de citation LLM.
- **Anthropic Blog** et **OpenAI Blog** : pour comprendre comment les moteurs évoluent et anticiper les changements de comportement.

- **r/SEO** sur Reddit : la communauté anglophone qui partage les changements détectés sur le terrain en temps réel.

8. Mot de la fin

Le SEO+GEO en 2026 n'est pas plus difficile que le SEO de 2020. Il est juste différent. Les fondamentaux changent peu (qualité du contenu, autorité, structure), mais les détails techniques et l'architecture des pages, eux, demandent une mise à niveau.

Ce que je vous ai partagé dans ce guide, c'est la version industrialisée et automatisée de ce que je faisais à la main il y a 18 mois. Le passage à l'industrialisation a pris quelques semaines de mise en place, mais le ROI est évident depuis la première semaine : plus de pages publiées, meilleure qualité moyenne, plus de citations LLM, plus de trafic, plus de leads.

Si vous appliquez ne serait-ce que 50% de ce qui est dans ce guide, vous serez devant 80% des sites français concurrents sur votre thématique. La majorité des acteurs en sont encore à se demander si le GEO est une mode passagère pendant que vous, vous capturez les premières citations.

Bonne mise en œuvre. Et si vous avez des questions, vous savez où me trouver.

Lucas Fonseque
Consultant SEO & IA — Toulouse
Avril 2026

Lucas Fonseque

Consultant SEO & IA · Toulouse

Consultant SEO et IA basé à Toulouse, j'interviens en freelance auprès de clients qui veulent industrialiser leur présence digitale et tirer parti des LLM en 2026. Les méthodes de ce guide sont celles que j'utilise au quotidien, validées en production sur des centaines de pages.

MES PRESTATIONS

- **Audit SEO + GEO** — Diagnostic complet de votre présence Google + LLM, plan d'action 90 jours.
- **Mise en place de Skills Claude** — Installation clé en main adaptée à votre stack. 1h30 par page garanti.
- **Production éditoriale** — Articles SEO+GEO sur abonnement (40 € HT/article ou 300 € HT le pack de 10).
- **Formation et accompagnement** — Sessions individuelles ou en équipe sur le SEO+GEO 2026 et les LLM.

COMMENT ME CONTACTER

Site web	**lucasfonseque.fr**
Calendly	**30 minutes pour cadrer votre projet**
LinkedIn	**/in/lucas-fonseque**
YouTube	**@lucas-fonseque**
