MAGIPA

Playbook Claude Code

4 semaines pour transformer tes opérations avec l'IA.

Guide pratique pour dirigeants de TPE et PME

Checklists, templates, astuces concrètes à chaque étape

Aucune compétence technique requise

MAGIPA Consulting — Saint-Pierre, La Réunion — 2026
martin@magipa.fr

Sommaire

01 **Avant de commencer** — Pré-requis et état d'esprit

02 **Semaine 1 — Foundation** — Poser les bases, créer ton premier skill

03 **Semaine 2 — Context Layer** — Rendre Claude intelligent et contextuel

04 **Semaine 3 — Quality Gates** — Sécuriser, chaîner, automatiser

05 **Semaine 4 — Compounding** — Le système s'améliore seul

06 **Templates de Skills** — 4 skills prêts à copier-coller

07 **Les 10 anti-patterns** — Les erreurs à éviter absolument

08 **Ressources** — Communautés, docs, repos GitHub

01 — AVANT DE COMMENCER

Pré-requis et état d'esprit

Ce playbook est conçu pour des dirigeants de TPE et PME qui veulent utiliser Claude Code pour automatiser leurs tâches récurrentes. Tu n'as besoin d'aucune compétence technique. Si tu sais écrire un e-mail structuré, tu peux écrire un skill.

Ce dont tu as besoin :

- Un abonnement Claude (Pro ou Team)
- Claude Code installé (demande à ton prestataire tech ou suis le guide Anthropic)
- 30 minutes par jour pendant 4 semaines
- Un carnet pour noter tes observations (physique ou numérique)

Les 3 principes fondateurs :

1. Un skill parfait > dix skills bancals. Ne te disperse pas. Un seul workflow bien rodé va te convaincre et te montrer la voie pour les suivants.

2. Fais-le d'abord à la main. Avant d'automatiser quoi que ce soit, fais la tâche toi-même 3 fois. Note les étapes, les exceptions, les cas tordus. C'est la matière première de ton skill.

3. Itère, ne vise pas la perfection. Ton premier skill sera imparfait. C'est normal. Après 5 itérations, il sera solide. Après 20 utilisations, il sera excellent.

■ Avant de démarrer
Prends 10 minutes maintenant pour lister les 5 tâches que tu fais le plus souvent dans ta semaine. Chronomètre chacune. La plus longue ET la plus répétitive sera ta première cible d'automatisation. Garde cette liste — tu vas y revenir souvent.

02 — SEMAINE 1

Foundation — Poser les bases

Objectif : un seul skill qui fonctionne à la perfection.

Jour 1-2	**Lance /init et configure CLAUDE.md** C'est la commande fondatrice. Elle génère toute l'arborescence de fichiers (.claude/, CLAUDE.md, etc.). Ouvre ensuite CLAUDE.md et écris 5 à 10 règles qui te définissent : ton métier, ta façon de parler aux clients, tes contraintes principales. Ces règles seront respectées à chaque session sans que tu les rappelles.
Jour 2-3	**Identifie tes 3 tâches les plus répétitives** Prends un carnet. Note les 3 tâches que tu fais toutes les semaines avec les mêmes étapes : rédiger un brief, faire un rapport, qualifier un prospect, répondre à un appel d'offres. Chronomètre-toi sur chacune. La plus longue et régulière sera ta première cible.
Jour 3-4	**Crée ton premier skill à la main** Choisis la tâche la plus simple et la plus fréquente. Écris le skill en 5 blocs : frontmatter (paramètres), inputs (données nécessaires), steps (étapes numérotées), output (format de sortie), definition of done (conditions de clôture). Pas besoin d'être parfait — juste fonctionnel.
Jour 4-5	**Teste-le 5 fois, note chaque problème** À chaque test, note ce qui a cloché : une étape manquante, un format pas respecté, un cas particulier oublié. Ajoute une règle dans le skill pour chaque erreur. Après 5 itérations, ton skill tourne de façon fiable dans 90% des cas.

■ Le test du stagiaire

Avant de codifier ton skill, explique-le à voix haute comme si tu formais quelqu'un de nouveau. Si tu hésites, c'est qu'il manque une étape. Si tu dis 'ça dépend', c'est qu'il faut ajouter une condition. Ce test oral transforme un skill moyen en skill solide.

Exemple de CLAUDE.md minimal :

```
@import .claude/rules/style-communication.md @import .claude/rules/process-commercial.md ## Mon activité Je dirige une PME dans le secteur [X]. Clients : [Y]. Territoire : [Z]. Langue de travail : français. Ton : direct, professionnel, sans jargon. ## Comportements attendus - Propose toujours une version "rapide" et une "robuste" - Format livrable : titre clair + 3 points max + next step - Ne modifie jamais un fichier client sans confirmation - En fin de tâche : résume en 2 phrases ce qui a été fait
```

Checklist fin de semaine 1

- ■ J'ai lancé /init et configuré mon CLAUDE.md avec 5+ règles personnelles
- ■ J'ai identifié et chronométré mes 3 tâches les plus répétitives
- ■ J'ai créé mon premier skill (frontmatter + inputs + steps + output + done)

- J'ai testé 5 fois et corrigé à chaque itération
- Mon skill tourne seul sur un cas réel sans intervention de ma part

03 — SEMAINE 2

Context Layer — Rendre Claude intelligent

Tes skills deviennent contextuels. Claude sait qui sont tes clients, tes standards, tes outils.

Jour 1-2	**Organise la hiérarchie mémoire** CLAUDE.md racine = tes préférences universelles (ton, langue, habitudes). Dans chaque dossier client, un CLAUDE.md local avec les spécificités de ce projet : interlocuteurs, stack, contraintes. Claude charge automatiquement le bon contexte selon où tu travailles.
Jour 2-3	**Un fichier = un sujet (modularité)** Crée .claude/rules/style-communication.md, process-commercial.md, clients-actifs.md. Commence par un seul. Cette modularité rend le système maintenable : tu modifies le style de communication sans toucher aux process commerciaux. Chaque fichier est réutilisable.
Jour 3-4	**Utilise les @imports** CLAUDE.md reste court (le sommaire). Les fichiers de détail sont chargés via @import. Moins de tokens consommés = Claude plus rapide, plus précis, moins cher par session. Règle absolue : CLAUDE.md < 150 instructions.
Jour 4-5	**Connecte ton premier MCP** Choisis l'outil que tu utilises le plus : Notion, Google Drive, HubSpot, Gmail. Un MCP connecté signifie que Claude peut lire ET écrire directement dedans. 'Crée la fiche client Dupont dans HubSpot' fonctionne en une commande. Zéro copier-coller.

■ Le test du nouveau projet

Crée un nouveau dossier client, lance Claude dessus, et vérifie qu'il charge automatiquement le bon contexte. S'il te demande 'quelle est ton activité ?', ta hiérarchie n'est pas bien configurée. Le but : zéro briefing à chaque démarrage de session.

Checklist fin de semaine 2

- ■ CLAUDE.md racine configuré avec mes préférences universelles
- ■ Au moins 1 fichier de rules thématique créé dans .claude/rules/
- ■ @imports fonctionnels dans mon CLAUDE.md principal
- ■ Un MCP connecté et testé (Notion, Drive, HubSpot ou Gmail)
- ■ Mon skill de semaine 1 fonctionne avec le contexte enrichi

04 — SEMAINE 3

Quality Gates — Sécuriser et chaîner

Tu mets des filets de sécurité et tu chaînes tes premiers workflows.

Jour 1-2	**Ajoute des post-hooks** Après chaque génération de document, Claude auto-formate selon tes standards (nommage, structure, ton) et envoie un résumé dans Notion ou Slack. Tu ne vérifies plus le formatage — chaque livrable sort conforme automatiquement. Un gain invisible mais massif.
Jour 2-3	**Installe tes gardes-fous (pre-hooks)** Claude ne peut pas : supprimer un fichier client, modifier des données financières, toucher à la prod sans ta validation explicite. Avec disable-model-invocation, tu contrôles quels skills Claude peut lancer seul. Tu dors tranquille.
Jour 3-4	**Chaîne deux skills ensemble** Exemple : /qualifier-prospect récupère les infos mail, crée la fiche CRM, génère le brief, planifie le call. Du premier contact au démarrage projet en une seule commande. Ce qui prenait 3h se fait en 5 minutes — et sans oubli.
Jour 4-5	**Adapte un skill existant depuis la communauté** Va sur skillhub.club : 7000+ skills créés par d'autres dirigeants et développeurs. Trouve un pattern proche de ton besoin, copie-le, adapte-le. Tu économises des heures de tâtonnement en partant d'un skill déjà éprouvé.

■ Le garde-fou du vendredi

Chaque vendredi, revois les 5 dernières actions que Claude a exécutées dans la semaine. Demande-toi : 'Y en a-t-il une qui aurait pu causer un problème ?'. Si oui, ajoute un pre-hook de sécurité. Cette habitude hebdomadaire rend ton système bullet-proof en quelques semaines.

Checklist fin de semaine 3

- ■ Un post-hook fonctionnel (auto-format + notification)
- ■ Au moins 1 pre-hook de sécurité configuré
- ■ Deux skills chaînés en un mini-workflow fonctionnel
- ■ Un skill adapté depuis skillhub.club ou la communauté
- ■ disable-model-invocation activé sur les skills sensibles

05 — SEMAINE 4

Compounding — Le système apprend seul

Tout devient exponentiel. Tu travailles moins qu'en semaine 1 — et tu produis davantage.

Jour 1-2	**Crée un skill /learn** Après chaque tâche terminée, Claude se pose 3 questions automatiquement : qu'est-ce qui a marché ? Qu'est-ce qui a raté ? Quelles nouvelles règles ajouter ? Il génère max 5 points d'amélioration et les ajoute à la mémoire. Le système grandit sans que tu interviennes.
Jour 2-3	**Intègre un /review systématique** Avant chaque envoi client : Claude vérifie l'exactitude des faits, la complétude du livrable, le respect du ton, le formatage. Tu gardes le contrôle qualité sans vérifier chaque ligne. Tu valides le résultat — tu n'exécutes plus le contrôle.
Jour 3-4	**Duplique et adapte (Scale)** Le skill commercial qui marche ? Adapte-le pour le recrutement en changeant les inputs et le template de sortie. Chaque nouvelle itération te coûte 10x moins d'effort que la première. En 5 minutes tu as un nouveau workflow opérationnel.
Jour 4-5	**Configure ton premier subagent** Un agent avec sa propre mémoire et ses propres skills : ton commercial IA, ton analyste, ton assistant admin. Chacun a son contexte dédié, ses règles, ses accès MCP. Après un mois, il connaît tes patterns mieux qu'un nouveau collaborateur.

■ Le journal de bord IA

Crée un fichier .claude/learnings.md que Claude met à jour automatiquement. Chaque semaine, relis les 5 derniers apprentissages. Tu verras le système devenir plus intelligent semaine après semaine — et tu découvriras des patterns que tu n'aurais jamais vus toi-même.

Checklist fin de semaine 4

- ■ Skill /learn configuré et fonctionnel après chaque tâche
- ■ /review actif avant chaque livraison client
- ■ Au moins 2 skills dupliqués et adaptés à d'autres contextes
- ■ Un subagent spécialisé configuré avec sa propre mémoire
- ■ Mon fichier learnings.md contient au moins 10 apprentissages

4 Skills prêts à copier-coller

/brief-projet

```
--- name: brief-projet description: Génère un brief projet complet et actionnable. À invoquer après un premier échange client. allowed-tools: Read, Write, mcp__notion__create_page model: claude-sonnet-4-5 --- ## Inputs requis - Nom du client + secteur d'activité - Objectif principal du projet (1 phrase) - Budget approximatif et deadline - Contraintes connues ## Steps 1. Reformuler l'objectif : "Tu veux X pour Y d'ici Z" — valider 2. Identifier les 3 risques principaux 3. Proposer 2 approches (rapide vs robuste) 4. Générer le brief : contexte > enjeux > livrables > jalons 5. Self-check : le client peut-il valider sans appel ? ## Definition of done - [ ] Objectif sans ambiguïté - [ ] Livrables avec dates et responsables - [ ] Fichier créé : CLIENT_DATE_brief.md
```

/rapport-mensuel

```
--- name: rapport-mensuel description: Génère le rapport mensuel d'activité pour un client. À lancer en fin de mois. allowed-tools: Read, Write, mcp__notion__search, mcp__hubspot__get_contact model: claude-sonnet-4-5 --- ## Inputs requis - Nom du client - Période couverte (mois/année) - Métriques clés à suivre (si pas définis dans le contexte client) ## Steps 1. Récupérer les données du mois dans Notion/HubSpot 2. Consolider les métriques clés en tableau 3. Identifier les 3 faits marquants du mois 4. Rédiger les recommandations pour le mois suivant 5. Formater selon le template MAGIPA ## Definition of done - [ ] Données vérifiées et sourcées - [ ] Maximum 2 pages, pas de jargon technique - [ ] Fichier : CLIENT_MOIS_rapport.md
```

/qualifier-lead

```
--- name: qualifier-lead description: Analyse un prospect avant le premier appel. Évalue le potentiel et prépare les questions. allowed-tools: Read, WebSearch, mcp__hubspot__get_contact model: claude-sonnet-4-5 --- ## Inputs requis - Nom de l'entreprise - Source du contact (site web, recommandation, salon, etc.) - Premier message reçu (si disponible) ## Steps 1. Rechercher l'entreprise : secteur, taille, actualités récentes 2. Évaluer le potentiel sur 3 critères : budget estimé, urgence, fit avec nos offres 3. Identifier 5 questions à poser lors du premier appel 4. Préparer un pitch personnalisé de 30 secondes 5. Générer la fiche prospect structurée ## Definition of done - [ ] Fiche prospect complète avec score de qualification - [ ] 5 questions prêtes pour l'appel - [ ] Pitch personnalisé écrit
```

/propale

```
--- name: propale description: Génère une proposition commerciale personnalisée à partir du brief projet. allowed-tools: Read, Write, mcp__notion__search model: claude-sonnet-4-5 disable-model-invocation: true --- ## Inputs requis - Brief projet validé (ou lien vers le fichier) - Tarification applicable - Conditions particulières (remise, paiement échelonné, etc.) ## Steps 1. Relire le brief projet et extraire les enjeux clés 2. Structurer la propale : contexte > approche > livrables > planning > tarif 3. Rédiger avec ton professionnel, sans jargon, orienté bénéfices client 4. Ajouter les conditions générales et mentions légales 5. Self-check : le client comprend-il la valeur sans explication orale ? ## Definition of done - [ ] Propale autonome (pas besoin d'appel pour comprendre) - [ ] Tarifs clairs avec détail par phase - [ ] Fichier : CLIENT_DATE_propale.md
```

Les 10 erreurs à éviter

Erreur	Exemple	Solution
Tout mettre dans CLAUDE.md	*Un fichier de 400 lignes = Claude hallucine et oublie des règles.*	`Utilise @imports. CLAUDE.md < 150 instructions.`
Automatiser avant de faire soi-même	*Tu automatises tes erreurs sans t'en rendre compte.*	`Fais la tâche 3x à la main d'abord.`
Ignorer les 7000 skills existants	*Tu réinventes la roue pendant 4h.*	`10 min sur skillhub.club avant de créer.`
Pas de garde-fous	*Claude supprime un dossier client par accident.*	`Pre-hooks + disable-model-invocation.`
Pas de context fork	*Après un long output, Claude oublie tes règles.*	`Fork le contexte pour les outputs longs.`
Règles critiques seulement en CLAUDE.md	*Oublié 1 fois sur 3 car dépend de la lecture.*	`Ce qui doit arriver CHAQUE fois = Hooks.`
Ne pas tester suffisamment	*Le skill échoue sur un vrai cas client.*	`5 tests minimum sur des cas variés.`
Créer des skills sans les utiliser	*8 skills créés, 2 utilisés, 6 oubliés.*	`1 skill à la fois. Utilise-le 10x d'abord.`
Hyper-systématiser sans agir	*3 jours à organiser, 0 skill qui tourne.*	`Un skill imparfait qui tourne > architecture parfaite sans usage.`
Ne pas adapter les skills	*Le skill dérive après 6 semaines sans mise à jour.*	`Kaizen : chaque vendredi, 5 min de revue.`

08 — RESSOURCES

Tout ce dont tu as besoin

Communautés de Skills

- **skillhub.club** — 7 000+ skills évalués par IA
- **ayclaude.com** — Community curated
- **agent37.com/skills** — Reddit community patterns
- **awesomeclaude.ai** — Ressources curées
- **skillsmp.com** — Agent marketplace
- **skills.sh** — Annuaire searchable

Documentation officielle Anthropic

- docs.anthropic.com/claude-code/skills
- docs.anthropic.com/claude-code/hooks
- docs.anthropic.com/claude-code/memory
- docs.anthropic.com/claude-code/sub-agents

GitHub — Les repos essentiels

- **awesome-claude-code** — La bible communautaire
- **awesome-agent-skills** — 200+ skills documentés
- **second-brain-skills** — Exocortex et second cerveau
- **anthropics/skills** — Repo officiel Anthropic

Tu veux aller plus vite ?
MAGIPA configure Claude Code directement pour ton activité. On analyse tes process, on construit tes Skills sur-mesure, on connecte tes outils. En 4 semaines, un système opérationnel — sans passer par la case 'apprendre à coder'.

Contact : martin@magipa.fr
MAGIPA Consulting — Saint-Pierre, La Réunion