

• EDIÇÃO REVISADA 2026

GAUNTLET LOOP GUIA PRÁTICO

Edição 2 - 2026

Como usar agentes construtores e críticos para elevar a qualidade de projetos, pesquisas, análises e decisões. **Uma arquitetura de controle de qualidade, não um revisor melhor.**

▶ COMEÇAR LEITURA →

G GEARIA • MÉTODO VALIDADO

4

AGENTES
ESPECIALIZADOS

100%

COMPARAÇÃO
CEGA

0

AUTO-APROVAÇÃO

1. O que é o Gauntlet Loop

O Gauntlet Loop é uma **arquitetura de controle de qualidade** que separa responsabilidades entre quem define o objetivo, quem constrói e quem critica. Não é um agente produzindo e aprovando o próprio trabalho. É comparação com referência externa e registro auditável.

CICLO GAUNTLET — FLUXO OFICIAL



Lógica de decisão

FALSO Corrigir e repetir. Nossa busca é por **repetição consistente**, não por aprovação única.

SIM Aprovar, registrar evidência e avançar para próxima etapa.

PRINCÍPIO DE OURO

Quem constrói nunca dá o veredito final. Separação total de contexto entre Builder e Critic.

Isolamento obrigatório via </gauntlet/>

A diferença entre rubrica e barra

Dimensão	Rubrica (subjetiva)	Barra (auditável)
Origem	Criada pelo Builder	Externa e pré-aprovada
Exemplo	'texto bom'	Relatório aprovado diretoria XYZ, nota 9.2
Falha típica	Auto-preferência	Não encontrada
Ação	Reescrever prompt	Substituir barra

2. Como escolher uma barra confiável

Barra é a referência externa contra a qual o Critic compara. Sem barra, não existe Gauntlet. A barra precisa ser **auditada, específica e anterior ao trabalho**. Exemplo: relatório de proposta comercial que já foi aprovado pela diretoria com nota explícita.

CHECKLIST DA BARRA CONFIÁVEL

- ✓ Existe antes do Builder? (não pode ser gerada junto)
- ✓ Foi aprovada por humano especialista externo?
- ✓ Tem critérios binários: passou / falhou?
- ✓ É específica do domínio? (não genérica)
- ✓ Pode ser citada como evidência em auditoria?
- ✓ Não contém instruções que o Builder usou?

Nota de atenção: O cansaço com filtro de preparação deve lembrar que método colaborativo exige objetividade. Se a barra não tem detalhe citável (página, métrica, assinatura), não é barra.

TABELA: OBJETIVO — CLASSIFICAÇÃO DA BARRA

Empresa

Proposta aprovada, contrato fechado, NPS >8

Sabor esmiuçado

Acadêmico

Artigo aceito / revisado por pares

Aceltado

Dados

Dataset com ground truth, métricas AUC>0.9

Sinais extremos

Decisão

Alta com responsáveis + resultado mensurável

Decisão registrada

Código

Teste e2e passando em main

Build verde

3. Os agentes e suas responsabilidades

1 OBJETIVO → 2 BARRA RE → 3 BUILDER → 4 CRITIC



Diagrama oficial — 4 agentes + isolamento

Lead define objetivo e barra. Builder constrói sem ver critic. Critic compara cegamente com barra. Validator executa testes. Nenhum agente lê prompt do outro.

LEAD

Isolado

Define objetivo, público, resultado esperado. Escolhe e trava a barra. Não constrói.

BUILDER

Isolado

Constrói artefato. Nunca recebe barra completa, apenas objetivo. Pode iterar com Validator.

CRITIC

Isolado

Compara Builder vs Barra sem saber quem é quem. Só responde PASSOU / FALHOU / BLOQUEADO + evidências.

VALIDATOR

Isolado

Validação executável: testes, lints, links, schema. Sem opinião, só binário.

Regra de ouro: Quem constrói não dá veredito final. Se Builder e Critic compartilham contexto, o loop está quebrado.

4. Quando usar / não usar + limites da comparação cega

TESTE DE ELEGIBILIDADE

- ✓ Existe barra auditável anterior?
- ✓ Resultado é verificável binário?
- ✗ Tarefa é criativa-pura sem critério?
- ✗ Você quer só-brainstorm?
- ✓ Há custo de erro alto se falhar?

Se 2+ respostas forem "não" na coluna esquerda, Gauntlet não é a ferramenta. Use revisão humana simples.

PROTOCOLO DE COMPARAÇÃO REFORÇADA — 8 PASSOS

- | | |
|---------------------|---------------------------|
| 1. Travar barra | 2. Anonimizar Builder |
| 3. Critic cego | 4. Evidência página/linha |
| 5. Validator exec | 6. Registro PASSOU/FALHOU |
| 7. Correção Builder | 8. Repetir até SIM |

NÍVEIS DE RIGOR

Básico Intermediário Reforçado Crítico

⚠ VIESES QUE CEGUEIRA NÃO REMOVE

Auto-preferência de modelo, viés de comprimento (prefere texto longo), viés de formatação bonita, e confirmação. Mitigue com Validator executável + evidência obrigatória (página, linha, commit).

Validator + Isolamento de contexto

CRITIC VS VALIDATOR

Dimensão	Critic	Validator
Tipo	Julgamento com referência	Execução binária
Exige barra?	Sim, auditável	Não, mas recomenda
Saída	PASSOU/FALHOU + evidência	PASSOU/FALHOU/BLOQUEADO + log
Pode opinar?	Sim, com evidência	Nunca

Contrato de saída — JSON obrigatório

```
{
  "veredito": "PASSOU | FALHOU | BLOQUEADO",
  "evidencias": ["p12, linha 4: valor diverge da barra"],
  "comparacao": "barra vs builder lado a lado",
  "proximo_passo": "corrigir X"
}
```

CHECKLIST VALIDATOR CONFIÁVEL (8 ITENS)

- ✓ Roda sem LLM (script puro)
- ✓ Retorna exit code 0/1
- ✓ Gera log auditável
- ✓ Não escreve no artefato
- ✓ Valida schema, links, testes
- ✓ Timeout definido
- ✓ Isolado em /gauntlet/validator/
- ✓ Sem dependência de prompt do Builder

ISOLAMENTO — O QUE CRITIC PODE/NÃO PODE RECEBER

PODE

Barra, Builder output
anonimizado, instruções de
comparação

NUNCA

Prompt do Builder, histórico,
nome do modelo, auto-
avaliação

```
/gauntlet/
├─ lead/ objetivo.md + barra.md
├─ builder/ output.md
├─ critic/ veredito.json
└─ validator/ run.sh + logs/
```

5. Instalação e Execução

PRÉ-REQUISITOS

Claude Code ou Codex, pasta /gauntlet/ versionada, barra travada em .md, Node 18+ para Validator opcional.

Checklist de definição

- ✓ [ARTEFATO] definido
- ✓ [PÚBLICO] definido
- ✓ [RESULTADO] mensurável
- ✓ Barra auditável anexada
- ✓ Critério de parada escrito

INSTALAÇÃO CLAUDE CODE

```
# Windows (PowerShell)
irm https://claude.ai/install.ps1 | iex

# macOS/Linux
curl -fsSL https://claude.ai/install.sh | sh

# Validar
claude --version
```

INSTALAÇÃO CODEX (OPENAI)

```
npm i -g @openai/codex
codex auth login
codex --help
```

COMO EXECUTAR GAUNTLET DO INÍCIO AO FIM — 5 ETAPAS

1

1. Lead trava objetivo + barra

Escreva /gauntlet/lead/objetivo.md com [ARTEFATO], [PÚBLICO], [RESULTADO] e /gauntlet/lead/barra.md com evidência citável. Trave commit.

2

2. Builder constrói isolado

Roda só com objetivo, sem barra. Output em /gauntlet/builder/output.md. Pode usar Validator para testes locais.

3

3. Validator executa

Roda /gauntlet/validator/run.sh. Se BLOQUEADO, Builder corrige antes do Critic.

4

4. Critic compara cegamente

Recebe barra + builder anonimizado. Retorna veredito.json com PASSOU/FALHOU + evidências página/linha.

5

5. Loop até SIM

Se FALHOU, Builder corrige com base só no veredito (não na barra). Repetir. Registrar log auditável.



Dica GearIA: Use `/gauntlet-loop` como skill. Ela já cria pastas e contratos.

6. Template mestre e Orçamento

TEMPLATE PARAMETRIZADO — COPIAR E PREENCHER

Objetivo Gauntlet

[ARTEFATO]: ex: proposta comercial de 5 páginas

[PÚBLICO]: ex: diretoria de TI de varejo, 1000+ lojas

[RESULTADO]: ex: aprovação sem retrabalho + assinatura em 48h

Barra travada

Fonte: /docs/proposta_aprovada_XYZ_2025.pdf p.12-18, nota 9.2 diretoria

Crêterios binários: [] contém ROI calculado [] prazo explícito [] risco mapeado

Critério de parada

Parar quando Critic = PASSOU 2x seguidas ou 5 ciclos.

[ARTEFATO]

[PÚBLICO]

[RESULTADO]

Circuito de segurança: Se Builder pedir barra, Lead responde BLOQUEADO. Barra nunca vaza para Builder.

ORÇAMENTO, PARADA E ESCALONAMENTO

Criticidade	Ciclos máximos
Baixa — rascunho interno	3 ciclos
Média — cliente externo	5 ciclos
Alta — contrato/decisão	7 ciclos + humano
Critica — compliance/legal	3 ciclos + revisão humana obrigatória

Gatilhos de escalonamento (7 itens)

- ⚠ Critic FALHOU 3x no mesmo ponto
- ⚠ Validator BLOQUEADO 2x
- ⚠ Builder pede esclarecimento de barra
- ⚠ Tempo > 2x estimado
- ⚠ Divergência entre 2 Critics
- ⚠ Barra desatualizada >6 meses
- ⚠ Risco legal/financeiro detectado

7. Quatro aplicações práticas



Empresas

Propostas, relatórios, contratos

Template cogorido



Academia

Artigos, revisões, teses

Usado por revisores



Dados

Análises, dashboards, modelos

Ground truth auditado



Decisões

Estratégia, priorização, OKRs

Ata + resultado

CASO DETALHADO — EMPRESA: REDUZIR RETRABALHO DE PROPOSTAS COMERCIAIS

Configuração	Valor Gauntlet
[ARTEFATO]	Proposta 5p: contexto, solução, ROI, prazo, riscos
[PÚBLICO]	Diretoria TI varejo 1000+ lojas, ticket 300k
[RESULTADO]	Aprovação sem retrabalho, assinatura 48h
Barra	Proposta Aprovada XYZ 2025 p.12-18 nota 9.2
Validator	Check: ROI calculado? Prazo explícito? 5 seções?
Parada	PASSOU 2x ou 5 ciclos

Resultado observado CearIA: retrabalho caiu de 2.3 versões para 0.4 por proposta. Tempo médio de aprovação de 6.2 dias para 1.8 dias.

OUTRAS APLICAÇÕES EMPRESARIAIS

- SOWs e escopos técnicos
- Relatórios de due diligence
- One-pagers de produto
- Respostas RFP / editais
- Documentação técnica para auditoria

Checklist de implementação

- ✓ Mapear 3 barras aprovadas
- ✓ Criar /gauntlet/ no repo
- ✓ Rodar 1 ciclo piloto
- ✓ Medir FALHOU vs PASSOU

8. Métricas e Solução de problemas

MÉTRICAS DE EFICÁCIA DO MÉTODO

Métrica	O que mede	Interpretação
Taxa PASSOU 1º ciclo	Qualidade inicial Builder	<40% = melhorar prompt Lead
Ciclos até PASSOU	Esforço total	Ideal 1-2, >4 = barra fraca
FALHOU recorrente	Ponto cego	Mesmo motivo 3x = revisar barra
BLOQUEADO Validator	Erro executável	>20% = Builder ignorando Validator
Tempo / ciclo	Custo operacional	Meta <12 min com Claude Code

Painel mínimo mensal (copiar para Notion/Sheets)

Total ciclos

47

PASSOU 1º

62%

BLOQUEADO

8%

SOLUÇÃO DE PROBLEMAS

Problema	Causa	Correção
Skill não aparece	/gauntlet/ fora do repo	Mover para raíz e commit
Critic aprova tudo	Barra genérica	Trocar por barra específica com nota
Loop infinito	Sem critério de parada	Definir max 5 ciclos
Validator não roda	Permissão run.sh	chmod +x + timeout 30s
Builder vê barra	Contexto vazio	Isolar pastas, limpar histórico

Antipadrão fatal: pedir para o mesmo agente "revisar melhor". Isso não é Gauntlet, é auto-preferência. Sempre separe.

9. Configurações rápidas + Conclusão

Empresa — proposta

```
/gauntlet-loop artefato=proposta Sp público=diretoria TI resultado=assinatura
48h barra=.docs/proposta_aprovada.pdf
```

Estudo acadêmico

```
/gauntlet-loop artefato=revisão sistemática público=banca doutorado
resultado=aprovação sem ressalvas barra=artigo aceito Qualis A1
```

Análise de dados

```
/gauntlet-loop artefato=dashboard churn público=C-level resultado=decisão de
intervenção barra=dashboard aprovado Q3 com AUC 0.91
```

Decisão estratégica

```
/gauntlet-loop artefato=one-pager priorização público=founders resultado=OKR
definido barra=ata decisão anterior com resultado mensurável
```

CONCLUSÃO

Gauntlet Loop **não é pedir para a IA revisar melhor**. É desenhar um sistema onde qualidade é verificável, não opinativa.

Quando você separa quem constrói de quem julga, trava uma barra auditável anterior e obriga evidência de página/linha, você elimina 3 falhas comuns: auto-preferência, alucinação de qualidade e “parece bom” sem critério.

O método funciona porque transfere a confiança de um prompt esperto para uma arquitetura auditável.



Comece pequeno, mas comece travado.

Escolha 1 artefato, 1 barra aprovada, 1 critério de parada de 3 ciclos. Rode 1 vez. Meça. Depois escale. O valor do Gauntlet não está no tamanho, está na trava.

CHECKLIST FINAL DE IMPLEMENTAÇÃO

- ✓ Barra existe antes e é citável
- ✓ Pasta /gauntlet/ versionada
- ✓ Builder não recebe barra
- ✓ Critic exige evidência página/linha
- ✓ Validator executa e bloqueia
- ✓ Contrato PASSOU/FALHOU/BLOQUEADO
- ✓ Log auditável salvo
- ✓ Critério de parada definido

10. Referências e Links

SUMMARY — O QUE VOCÊ LEVA DESTA EDIÇÃO

1. Gauntlet é arquitetura, não prompt. Separe Lead/Builder/Critic/Validator.
2. Barra auditável é obrigatória. Sem barra, sem Gauntlet.
3. Comparação cega + evidência remove auto-preferência.
4. Validator binário é seu circuito de segurança.
5. Comece com 1 artefato, 3 ciclos, 1 barra aprovada.

REFERÊNCIAS COMENTADAS

Categoria	Fonte	Por que consultar
Método	RoboNuggets — Gauntlet Loop original	Definição de barra e isolamento
Crítica	Matt Shumer — Critic vs Builder	Como Critic cego supera self-refine
Agentes	Anthropic — Claude Code docs	Isolamento de contexto e skills
Agentes	OpenAI — Codex	Validator e execução
Viés	Self-Preference Bias (Anthropic 2024)	Prova de auto-preferência de LLM
Validação	GearIA — Validator executável	Checklists e run.sh modelo

LINKS OFICIAIS

[Claude Code Install](#)[Anthropic Docs — Agents](#)[OpenAI Codex](#)[GearIA — Método Gauntlet](#)

G **GearIA — Inteligência Aplicada**
Consultoria que implementa IA em processos reais de empresas. Método Gauntlet Loop validado em 40+ projetos de propostas, análises e decisões. Edição 2 revisada em 2026 com templates de produção.

contato@gearia.com.br[Edição 2 — 2026](#)

Licença de uso: Este guia é material interno GearIA para clientes. Reprodução parcial permitida com crédito. Templates /gauntlet/ podem ser copiados livremente para repositórios privados.