Perfeito. Abaixo está a **DOCUMENTAÇÃO PADRÃO (CANÔNICA) "PRONTO PARA IMPLEMENTAR"**, que vocês podem reutilizar em **todo projeto novo**.
Quando **todos** esses itens estiverem prontos, vocês podem iniciar a implementação com Claude Code sem se perder.

✅ PADRÃO "PRONTO PARA IMPLEMENTAR"

Checklist de Documentos Obrigatórios (para qualquer projeto)

> Objetivo: transformar ideia em software **sem deriva**, **sem confusão**, **sem scope creep**.

---

1) Identidade e Escopo

1.1 Documento de Visão (obrigatório)

- Propósito
- Problema que resolve
- O que é / o que não é
- Escopo e fora de escopo
- Princípios fundamentais

Saída: visão estável e não interpretável.

1.2 Constituição / Princípios Invioláveis (obrigatório)

- Valores permanentes
- Limites absolutos (o que nunca pode acontecer)
- Prioridades em conflito (ex: sobrevivência > velocidade)
- Regras de "não-deriva"

Saída: lista curta, clara e auditável.

1.3 Glossário Canônico (obrigatório)

- Definições fechadas de termos críticos
- Sinônimos proibidos
- Termos que não podem ser usados

Saída: linguagem única do projeto.

---

2) Comportamento do Sistema

2.1 Casos de Uso (obrigatório)

- Casos principais no formato padrão:
    - Contexto, objetivo, informações, incertezas, opções, riscos, decisão, consequência, aprendizado

Saída: comportamento esperado antes de qualquer técnica.

2.2 Fluxo Principal + Fluxos Alternativos (obrigatório)

Para 1–2 casos centrais:

- Fluxo ideal
- Fluxos alternativos
- Fluxos de exceção
- Fluxos de erro
- Fluxo abortado

Saída: estados e transições claros (sem tecnologia).

---

3) Contratos e Interfaces

3.1 Modelos Canônicos (obrigatório)

Lista fechada dos objetos e seus campos obrigatórios (conceitual → técnico):

- Entidades
- Campos
- Restrições
- O que é imutável vs editável

Saída: “modelo de dados” conceitual pronto para virar código.

3.2 Contratos de Entrada/Saída (obrigatório)

- Inputs do sistema (forms, eventos, requests)
- Outputs do sistema (decisões, registros, respostas)
- Regras de validação
- Erros esperados

Saída: API mental do sistema, mesmo sem stack.

3.3 Templates Oficiais (obrigatório)

- Template de input (ex: Solicitação)
- Template de output (ex: Contrato)
- Template de observação/registro
- Template de encerramento (se existir)

Saída: operação padronizada.

---

4) Regras Técnicas Mínimas (pré-implementação)

4.1 Especificação Técnica Mínima (obrigatório)

Documento curto com:

- Estados formais e transições permitidas
- Imutabilidade (append-only, nunca reescrever)
- Identificadores e rastreabilidade
- Regras de consistência
- Requisitos mínimos de auditoria

Saída: base para implementação sem divergência.

4.2 Critérios de Pronto (Definition of Done) (obrigatório)

- O que significa “feito”
- O que deve ter teste
- O que deve estar documentado
- O que deve estar demonstrável

Saída: evita “quase pronto”.

---

5) Planejamento de Entrega

5.1 Plano de Incrementos (obrigatório)

Dividir o sistema em 2–4 entregas:

- Incremento 1 (mínimo operacional)
- Incremento 2 (funcionalidade essencial)
- Incremento 3 (robustez)
- Incremento 4 (escala/opcional)

Saída: caminho executável sem tentar construir tudo.

5.2 Matriz de Escopo (obrigatório)

- Must / Should / Could / Won’t
- “Não fazer agora” explícito

Saída: trava de escopo.

---

6) Representação (opcional, mas recomendado)

6.1 Wireframes / UI estática (recomendado)

- Telas mínimas
- Campos obrigatórios
- Estados

Saída: alinhamento rápido e menos retrabalho.

✅ Resumo: Lista final “Pronto para Implementar”

Para começar Claude Code com segurança, tem que existir:

1. Documento de Visão
2. Constituição/Princípios invioláveis
3. Glossário canônico
4. Casos de uso completos
5. Fluxo principal + alternativos (1–2 casos)
6. Modelos canônicos (entidades + campos + regras)
7. Contratos de I/O + validações
8. Templates oficiais
9. Especificação técnica mínima (estados + imutabilidade + auditoria)
10. Definition of Done
11. Plano de incrementos
12. Matriz de escopo
13. (Recomendado) Wireframes/UI estática

Como isso vira “prompts para Claude Code”

Quando esses 12 itens estiverem prontos, aí sim você cria prompts em 3 camadas:

- Prompt do **incremento**
- Prompt do **módulo**
- Prompt do **arquivo**

Assim você controla o output e evita que o Claude “invente arquitetura”.

Se você quiser, eu já adapto essa lista **para o projeto Libervia ↔ Bazari**, marcando:
✅ já temos / ⏳ falta / ⚠️ precisa revisar.