

Índice

00 — Índice

“ Guia normativo para criar um **novo projeto web** no padrão Evológica/Curio. Consolidado a partir de investigação de código real, não de teoria — cada decisão aqui foi verificada contra uma implementação de verdade antes de virar regra. Não é tutorial de React. É o manual de padrões do time.

Como ler

Leia na ordem. Cada documento assume os anteriores.

#	Documento	Decide
01	Stack e decisões	O que é fixo, o que é negociável
02	Bootstrap	Do <code>git init</code> até a tela de login rodando
03	Estrutura de pastas	Onde cada arquivo mora
04	Ambientes e config	<code>config/*.json</code> , <code>public/config.json</code> , <code>.env</code>
05	Curio: conexão e sessão	Login, sessão, expiração, logout
06	Caso de uso	Como chamar o backend
07	React Query	Cache, erro global, query keys
08	Componentes	Regra dos 3 arquivos, catálogo <code>common/</code>
09	Layout e menu lateral	Shell, <code>menuTree</code> , abas

#	Documento	Decide
10	Rotas e proteção	<code>paths</code> / <code>routes</code> / guards
11	Formulários e validação	react-hook-form + zod
12	Tema e estilo	MUI, <code>sx</code> vs <code>styled</code>
13	Convenções	Nomenclatura, <code>_Prefixo</code> , PT-BR vs inglês
14	Qualidade e pré-commit	ESLint, Prettier, husky, lint-staged
15	Build e deploy	Builds por ambiente, ISAPI
16	Template de CLAUDE.md	Arquivo pronto para colar
17	Primeira tela	Receita end-to-end verificável
18	Harness	Disciplina de sessão entre devs
19	init.sh e Definition of Done	Verificação de baseline
20	Ciclo de desenvolvimento	Issue → branch → commits → MR → merge
21	Catálogo de hooks	Os 12 hooks/módulos reutilizáveis de <code>src/hooks/</code>
22	Testes E2E com Playwright	Setup do zero, login reutilizável, Page Object Model

Os documentos 00-19 seguem a numeração planejada em PROMPT.md. O **20**, **21** e **22** foram acrescentados depois — processo de trabalho, catálogo de hooks e testes E2E não estavam no plano original, e todos são pré-requisito para o time operar, não só para o código compilar.

Convenção destes documentos

- Todo snippet traz o caminho de destino no topo e é colável sem edição.
- Exemplos usam a entidade fictícia `Fornecedor`, não o domínio de nenhum projeto real.

Checklist — projeto novo pronto

Marque quando cada item estiver **verificado**, não quando parecer feito.

Fundação

- ☐ Repositório criado, `.node-version` com `24.19.0`, `fnm use` funcionando
- ☐ `npm run dev` sobe e a página carrega sem erro no console
- ☐ `npm run lint` passa com zero warnings
- ☐ `npx tsc --noEmit` passa
- ☐ Aliases (`@`, `@components`, ...) resolvem em **ambos** `vite.config.ts` e `tsconfig.json`
- ☐ Os 12 hooks/módulos de 21-CATALOGO-DE-HOOKS.md copiados, com `hooks/index.ts`

Configuração

- ☐ `config/dev.json` e `config/prod.json` existem e apontam para servidores reais
- ☐ `public/config.json` está no `.gitignore` (é gerado, não versionado)
- ☐ `.env.example` versionado; `.env` ignorado
- ☐ Nenhum token real commitado em `config/*.json`

Curio

- ☐ Login autentica contra o backend real
- ☐ Refresh da página mantém a sessão (reconexão por token)
- ☐ Sessão expirada desloga e redireciona para `/login` automaticamente
- ☐ Uma tela abre um caso de uso e recebe resposta do backend

Aplicação

- ☐ Menu lateral renderiza a partir do `menuTree`
- ☐ Rota protegida redireciona para login quando deslogado
- ☐ 404 funciona
- ☐ Erro de request aparece como notificação, sem `try/catch` na página

Qualidade

- ☐ `.husky/pre-commit` roda `lint-staged` e barra commit sujo
- ☐ `check-node-version.sh` barra Node errado
- ☐ Prettier formata no commit

Testes E2E

- ☐ `npx playwright test` roda a suíte com o setup de login funcionando (22-TESTES-E2E-PLAYWRIGHT.md)

Harness

- ☐ `harness/` criado com `guides/` e `_examples/` (`state/`, `handoffs/`, `user_preferences.example.md`)
- ☐ Só `harness/user_preferences.md` no `.gitignore` — resto do harness é versionado
- ☐ Seção de harness no `CLAUDE.md`, incluindo o passo de resolver `harness/<branch>/`
- ☐ `./init.sh` executado com sucesso (modo FAST)

Aviso sobre usar código existente como referência

Nenhuma implementação real é um exemplar perfeito. Antes de copiar de um projeto existente, confira a lista de anti-padrões recorrentes em 13-CONVENCOES.md — são inconsistências reais que já apareceram mais de uma vez e **não** devem ser replicadas.

Revision #3

Created Thu, Aug 20, 2026 4:39 PM by Geraldo Barbosa

Updated Tue, Aug 25, 2026 4:49 PM by Geraldo Barbosa