

01 — Stack e decisões

01 — Stack e decisões

“ Define a stack fixa do projeto web e o que cada peça resolve. Itens marcados **FIXO** não são escolha do projeto novo — mudá-los quebra compatibilidade com o time. Itens **NEGOCIÁVEL** podem variar se houver motivo declarado. Versões verificadas em 2026-08-19 contra o npm registry e testadas de verdade (`tsc --noEmit` , `lint` , `build` , tela no browser) — não é combinação só especulada.

Stack

Peça	Versão	Status	Por quê
Node	<code>>= 24</code>	FIXO	Última LTS ativa (Node 24, "Krypton"). <code>process.loadEnvFile</code> exige ≥ 20.6
TypeScript	<code>^6.0</code>	FIXO	Bumpado de 5.9. Teto real: <code>typescript-eslint</code> (v8, latest) exige <code>typescript >=4.8.4 <6.1.0</code> — <code>6.0.3</code> é a versão mais alta possível hoje. <code>7.x</code> fica pra quando houver suporte
Vite	<code>^8.2</code>	FIXO	Bumpado de 4. <code>@vitejs/plugin-react</code> v6 exige <code>vite ^8</code> — os três (<code>vite</code> , <code>@vitejs/plugin-react</code> , <code>vite-plugin-checker</code>) sobem juntos
React	<code>^19.2</code>	FIXO	Bumpado de 18. <code>@curio/client</code> só exige <code>react >=16.8.0</code> — sem bloqueio

Peça	Versão	Status	Por quê
<code>@curio/client</code>	<code>^1.4.2</code>	FIXO	Transporte proprietário para o backend. Não avaliado nesta rodada (decisão do projeto)
MUI (<code>@mui/material</code>)	<code>^9.3</code>	FIXO	Bumpado de 5 direto para 9 — ver migração MUI 5→9
<code>@mui/x-date-pickers</code>	<code>^9.11</code>	FIXO	Acoplado ao major do <code>@mui/material</code> . Peer cobre React 19
<code>@tanstack/react-query</code>	<code>^5.101</code>	FIXO	Bumpado de 4.36. <code>cacheTime</code> → <code>gcTime</code> já ajustado nos hooks
<code>react-router-dom</code>	<code>^7.18</code>	FIXO	Bumpado de 6. <code>BrowserRouter</code> não aceita mais a prop <code>future</code>
<code>react-hook-form</code>	<code>^7.85</code>	FIXO	Bumpado de 7.48 — exigido pelo <code>@hookform/resolvers</code> v5 (ver Zod abaixo)
<code>@hookform/resolvers</code>	<code>^5.9</code>	FIXO	Bumpado de 3 — obrigatório para usar Zod v4, não é opcional
<code>zod</code>	<code>^4.4</code>	FIXO	Bumpado de 3. Mudança de tipos exigiu ajuste em <code>useValidatedForm</code> — ver abaixo
<code>date-fns</code>	<code>^4.4</code>	FIXO	Bumpado de 2. Exige <code>@mui/x-date-pickers/AdapterDate</code> (v9 já é o adapter para v3/v4 — ver nota)
ESLint 10 + Prettier 3	—	FIXO	Bumpado de 8. Flat config (<code>eslint.config.js</code> , não mais <code>.eslintrc.json</code>) — ver 14
<code>typescript-eslint</code>	<code>^8.67</code>	FIXO	Bumpado de 7. Compatível com ESLint 8/9/10 e TS <code><6.1.0</code> — é o teto que bloqueia o TypeScript acima
<code>eslint-plugin-react-hooks</code>	<code>^7.1</code>	FIXO	Bumpado de 4. <code>recommended</code> ficou bem mais amplo (regras novas tipo <code>set-state-in-effect</code>) — ver 14

Peça	Versão	Status	Por quê
husky + lint-staged	—	FIXO	Gate de pré-commit. <code>lint-staged</code> bumpado de 15 para 17 (dev tool, sem mudança de config)
Emotion	<code>^11.14</code>	FIXO	Peer dependency do MUI

Sem suíte de testes por padrão. Este guia não inventa uma nem assume que ela existe. O gate de qualidade é type-check + lint + build — ver 14 e 19. Se o projeto novo quiser testes, essa é uma decisão a tomar explicitamente no início, não depois.

Por que nem tudo foi para o latest

Em 2026-08-19, depois de uma rodada completa de atualização testada de verdade, só sobrou um pacote fora do latest:

Pacote	Aqui	Latest hoje	Ficou de fora porque
TypeScript	<code>6.0.3</code>	7.0.2	<code>typescript-eslint@8.67.0</code> (latest estável) exige <code>typescript >=4.8.4 <6.1.0</code> — não há versão estável do <code>typescript-eslint</code> que aceite TS 7 ainda (só canary). <code>6.0.3</code> é o teto real: existe uma linha <code>6.x</code> entre o <code>5.9</code> antigo e o <code>7.0</code> novo, e ela cabe no range aceito — não pule direto de 5→7 sem checar se há uma minor intermediária. Bumpar além de <code>6.0.x</code> quebraria o lint, não é escolha.

Todo o resto (MUI, Vite, ESLint, date-fns, typescript-eslint, react-hooks) já está no latest — ver as notas de migração abaixo para quem for repetir isso num projeto mais antigo.

Migração MUI 5 → 9: o que mudou de verdade

O caminho **não é incremental por major** (5→6→7→8→9): `@mui/x-date-pickers@9` exige `@mui/material: "^7.3.0 || ^9.0.0"` — não aceita a v8 como peer. Suba os três pacotes (`@mui/material`, `@mui/icons-material`, `@mui/x-date-pickers`) juntos, direto pra v9.

Superfície de quebra real, testada com `tsc --noEmit`:

- **Sem uso de `Grid`** no projeto — a maior fonte de quebra entre majors do MUI não se aplicou aqui. Se o projeto novo usa `Grid`, verifique a API v1→v2 separadamente.
- **`InputProps`, `inputProps`, `InputLabelProps` foram removidos** de `TextField` / `Checkbox` (não são só deprecados — o tipo não aceita mais). Viram `slotProps`: `InputProps` → `slotProps.input`, `inputProps` → `slotProps.htmlInput`, `InputLabelProps` → `slotProps.inputLabel`. Isso afeta qualquer componente de `common/` que componha `TextField` (`FormField`, `CurrencyInput`, `PhoneInput`, `CnpjInput`, `DatePickerInput`, `DataTable`, `TransferListCard`).
- **Props de atalho de estilo direto em `Box` / `Stack` / `Typography` / `DialogTitle`** (`display`, `alignItems`, `justifyContent`, `fontWeight`, etc., sem passar por `sx`) pararam de tipar — MUI removeu o suporte a "system props" desses componentes. Mova pra `sx={{ ... }}`.
- **`disableEscapeKeyDown` foi removido** de `Dialog` / `Modal` sem substituto direto — se o `Dialog` não tem `onClose`, a prop já era redundante (nada fecha no Escape de qualquer forma); se tem, trate a `reason` dentro do próprio `onClose`.
- **`@mui/x-date-pickers` inverteu a convenção de adapter entre majors:** na v6/v7, `AdapterDateFns` era para date-fns v2 e `AdapterDateFnsV3` para v3/v4. Na v9, inverteu — `AdapterDateFns` (sem sufixo) já é o adapter para v3/v4, e `AdapterDateFnsV2` é o legado. **Confira o import de novo a cada major**, não assuma que o nome antigo continua correto.

date-fns e o adapter do x-date-pickers

Bumpar `date-fns` sem trocar o subpath do adapter (ver acima) quebra silenciosamente em runtime, não em tipo — o import resolve, mas o parsing de data se comporta diferente. Sempre os dois juntos.

Decisões que valem entender

O backend não é REST

`@curio/client` é um transporte RPC proprietário. Não existe `GET /api/fornecedores`. Existe: abrir um **caso de uso** por id numérico e enviar **requests nomeados** (`RM_OBTEM_LISTA`) dentro dele.

Consequência prática: não use `axios`, `fetch` direto (fora do carregamento do `config.json`), nem qualquer geração de cliente a partir de OpenAPI. Ver 06.

O `fetch` do config é a única exceção

`src/lib/curio/index.ts` faz `fetch("./config.json")` para descobrir a URL do serviço em runtime. Isso é intencional: permite trocar de ambiente sem rebuild. Ver 04.

React Query v5 — `gcTime`, não `cacheTime`

A opção `cacheTime` foi renomeada para `gcTime` na v5, em `defaultOptions.queries` e `defaultOptions.mutations` do `queryClient.ts`, e em qualquer `useQuery` que a declare (`useConnectQuery`). `isPending` já era o nome correto desde a v4.36 — nenhuma mudança ali.

Zod v4 exige `@hookform/resolvers` v5 e um ajuste de generics

Zod v4 mudou a arquitetura interna de tipos de `ZodType`. Duas consequências reais, encontradas rodando `tsc` de verdade, não hipotéticas:

1. `@hookform/resolvers` v3 só aceita Zod `^3` — subir o Zod **exige** subir os resolvers para v5 (que por sua vez exige `react-hook-form >= 7.55.0`). Não é uma escolha independente.
2. O hook genérico `useValidatedForm<T extends z.ZodType>` deixou de compilar: o `output` de `z.ZodType` em v4 não fecha automaticamente em `FieldValues`. A correção é dupla — apertar o bound do genérico para `z.ZodType<FieldValues>`, e no ponto de chamada do `zodResolver`, usar um cast de tipo (`as any` justificado + `as unknown as Resolver<...>`) porque o tipo interno `Zod4Type` do resolver não compõe com um schema genérico sem perder a inferência. Ver `src/hooks/useValidatedForm.ts` — o cast é só de tipo; `zodResolver` continua validando em runtime exatamente como antes.

React Router v7 — sem prop `future`

Os *future flags* (`v7_startTransition`, `v7_relativeSplatPath`) que o `BrowserRouter` aceitava na v6 viraram comportamento padrão na v7. A prop `future` não existe mais — remova-a de `main.tsx`.

Estado de servidor no React Query, estado de UI no componente

Não há Redux, Zustand ou store global. O que vem do backend vive no React Query; o resto é `useState` local ou Context (`AuthProvider` , `NotificationProvider` , `TabsContext`). Se você sentir falta de uma store global, provavelmente está guardando resposta de servidor no lugar errado.

Erro é global, não local

Nenhuma página trata erro de request com `try/catch` para exibir mensagem. O `QueryCache` / `MutationCache` do `queryClient` captura e empurra para o `NotificationProvider` . Ver 07.

Negociável

Item	Padrão sugerido	Quando mudar
Porta do dev server	<code>5000</code>	Conflito local; ajuste em <code>vite.config.ts</code>
Paleta do tema	Azul <code>#1976d2</code>	Identidade visual do produto novo
Navegação por abas	Opcional	Só adote se o produto realmente precisar de abas
Dark mode	Ausente	Se o produto exigir; ver 12
<code>sourcemap</code> em produção	<code>true</code>	Desligue se o bundle for público e sensível