



Arquitetura de observabilidade com Better Stack

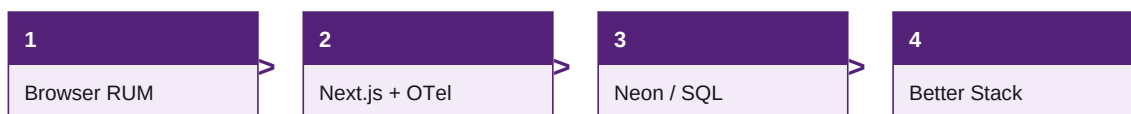
Visão unificada de experiência, aplicações, bancos de dados e infraestrutura com OpenTelemetry.

Arquitetura de referência

A Jovens Gênios adotará a Better Stack como plataforma central de observabilidade, mantendo OpenTelemetry como padrão neutro de instrumentação e transporte. A arquitetura reúne RUM, Web Vitals, erros, logs, métricas, traces, dashboards e alertas e preserva o contexto de uma interação desde o navegador até a operação executada no banco de dados.

O objetivo é ampliar a transversalidade da análise, reduzir o custo de ferramentas e armazenamento e aumentar a visibilidade entre os componentes da solução sem criar dependência de instrumentação proprietária.

Fluxo correlacionado da requisição



Origem	Sinal e transporte	Visibilidade na Better Stack
Navegador	RUM, Web Vitals, erros e replay; propagação W3C traceparent	experiência por rota normalizada, erro JavaScript e sessão relacionada
Next.js na Vercel	spans HTTP/servidor por OTLP e logs estruturados com trace_id e span_id	latência, status, serviço, dependências e logs da mesma requisição
Neon	span filho da chamada HTTP/SQL, métricas e logs PostgreSQL exportados por OTLP	operação sanitizada, latência SQL, conexões, erros e estado do PostgreSQL
ClickHouse Cloud	endpoint Prometheus coletado por um OpenTelemetry Collector	CPU, memória, consultas, concorrência, falhas e capacidade do serviço
AWS	CloudWatch e CloudTrail encaminhados pela integração da Better Stack	eventos operacionais, infraestrutura e auditoria no mesmo ambiente

Modelo de correlação

O cabeçalho W3C traceparent mantém a identidade da requisição entre navegador e servidor. O span da rota Next.js cria spans filhos para autenticação, serviços de aplicação e acesso ao Neon. Logs emitidos durante o processamento carregam os mesmos identificadores e são vinculados automaticamente ao trace correspondente.

Nível	Exemplo no waterfall	Atributos permitidos
Browser	rota ou ação do usuário	rota normalizada, duração, LCP, CLS, INP, tipo de erro
Servidor	POST /instituicoes/:id	http.route, status, duração, serviço e ambiente
Aplicação	autenticação e serviço	nome estável da operação, resultado e duração
Banco	SELECT institutions / Neon HTTP	db.system, db.operation.name, db.query.summary e duração
Log	evento relacionado ao span	trace_id, span_id, nível, serviço, mensagem sanitizada e timestamp

Cobertura de métricas e sinais

Necessidade	Sinal	Resultado esperado
Experiência	LCP, CLS e INP por rota	visibilidade da percepção real do usuário e dos Core Web Vitals atuais
Erros de frontend	exceções JavaScript e source maps	origem, versão, rota, frequência e sessão relacionada
Performance de rota	spans HTTP do Next.js	P50, P95, P99, taxa de erro e dependências lentas
Acesso a dados	spans CLIENT sanitizados para Neon	latência e tipo de operação sem parâmetros ou valores literais
PostgreSQL	métricas e logs nativos da Neon	conexões, erros, recursos e eventos do banco no mesmo painel
Analytics	métricas Prometheus do ClickHouse Cloud	capacidade, concorrência, uso de recursos e falhas de consulta
Investigação	logs, métricas e traces correlacionados	navegação direta da experiência ao componente causador

Política de retenção

A adoção da Better Stack incorpora a ampliação da retenção como parte da arquitetura. Os dados são separados por finalidade para combinar busca rápida, preservação forense e custo controlado. A plataforma permite configurar retenção por fonte e manter dados históricos em bucket compatível com S3 após a janela consultável.

Classe	Retenção definida	Forma de preservação
Logs operacionais e de aplicação	90 dias	consulta centralizada, filtros por serviço, rota, nível e trace
Eventos de segurança e auditoria	12 meses	período consultável e histórico preservado em armazenamento compatível com S3
Métricas e tendências de SLO	12 meses	séries agregadas para capacidade, disponibilidade e comparação histórica
Traces, RUM e session replay	30 dias	investigação de experiência e performance com amostragem controlada de replay

Controle de custo e proteção de dados

- Rotas normalizadas substituem URLs com identificadores e evitam cardinalidade desnecessária.
- Erros e Web Vitals são preservados integralmente; session replay utiliza amostragem controlada.
- Corpos de requisição, parâmetros SQL, valores literais, tokens, credenciais e dados pessoais não integram a telemetria.
- Spans de banco registram somente operações e resumos sanitizados, como `SELECT users`.
- Retenção por classe e armazenamento histórico de menor custo evitam manter todo sinal na camada de consulta rápida.
- Limites de ingestão e alertas de consumo tornam o custo observável e previsível.

Benefícios da arquitetura

Benefício	Efeito operacional
Análise transversal	uma requisição pode ser acompanhada do navegador ao SQL e aos sinais de infraestrutura
Maior visibilidade	logs, métricas, traces, RUM e alertas compartilham serviço, rota, ambiente e contexto
Redução de custos	uma plataforma central, OpenTelemetry, retenção por classe e histórico de menor custo
Menor tempo de diagnóstico	waterfalls e correlação automática eliminam pesquisas manuais entre ferramentas
Portabilidade	instrumentação OpenTelemetry reduz dependência de APIs proprietárias do fornecedor

Referências oficiais

[Better Stack - Distributed tracing e correlação](#)

[Better Stack - Configuração de fontes, retenção e bucket](#)

[Better Stack - Consulta de dados recentes e históricos](#)

[Better Stack - Integração AWS](#)

[Neon - Exportação de métricas e logs por OpenTelemetry](#)

[ClickHouse Cloud - Endpoint Prometheus](#)

[OpenTelemetry - Convenções de spans de banco](#)

[W3C - Trace Context](#)