

CAIO VILQUER CARVALHO

Desenvolvedor Full Stack | Java, Kotlin, TypeScript, React e Angular

São Paulo, SP · +55 (31) 98815-6118 · caio@vilquer.dev
linkedin.com/in/caio-vilquer · github.com/caiovilquer · vilquer.dev

Busca estágio em desenvolvimento de software ou vaga de desenvolvedor full stack júnior. Estudante de Engenharia de Computação na Poli-USP, com três sistemas construídos do modelo de dados à tela, dois em produção e em uso real. Escreve o backend em Java e Kotlin com Spring Boot e em TypeScript com NestJS, e as interfaces em React e Angular sobre o mesmo contrato de API, com testes automatizados, Docker e CI.

FORMAÇÃO

Bacharelado em Engenharia de Computação · Escola Politécnica da USP (Poli-USP) 02/2025 - 12/2029 (previsto)

COMPETÊNCIAS

- Linguagens:** Java, Kotlin, TypeScript e SQL
- Backend:** Spring Boot, NestJS e APIs REST
- Frontend:** React, Angular, TypeScript e Tailwind CSS
- Dados:** PostgreSQL, JPA/Hibernate e Prisma
- Engenharia:** testes automatizados, Docker, Git e CI/CD com GitHub Actions
- Idiomas:** inglês intermediário, leitura de documentação técnica e comunicação escrita

EXPERIÊNCIA

Desenvolvedor full stack · POLIATLETAS, Equipe de Atletismo da Poli-USP (atuação voluntária) 08/2025 - atual · em produção e uso real
NestJS · Fastify · PostgreSQL · Prisma · Redis · React · poliatletas.com.br

- Único responsável pelo sistema da equipe, do levantamento com treinadores e atletas ao deploy: mais de 100 atletas e membros da comissão técnica cadastrados e cerca de 50 em uso ativo, num sistema único para competições, resultados, recordes, rankings e treinos, no lugar do controle disperso em planilhas.
- Fez recordes pessoais, rankings e recordes da equipe saírem prontos e comparáveis entre provas de tempo, distância e pontuação, sem recálculo manual e sem tocar no histórico: unidade canônica e direção de comparação por prova.
- Reduziu o lançamento de uma competição inteira a um envio, sem duplicatas em caso de reenvio, e impediu que dois treinadores no mesmo treino apagassem o trabalho um do outro: importação em lote idempotente, conflito avisado em vez de sobrescrita, acesso separado por papel e auditoria das operações sensíveis.
- Entregou o ranking público que atleta e comissão consultam sem depender de planilha de ninguém, filtrável por prova e período e comparável entre temporadas: painéis em React com Tailwind CSS e Recharts, consumindo os tipos gerados do contrato da API.
- Sustenta a produção com mais de 250 testes, rankings em cache, health checks e migrações no deploy, publicando novas versões com o sistema no ar durante a temporada; o contrato OpenAPI gera os tipos do frontend, então mudança de API quebra no build, não no uso.

PROJETOS

ROTINA PET · Gestão de cuidados com assistente de IA e RAG 05/2025 - atual · em produção
Kotlin · Spring Boot · Angular 20 · Angular Material · PostgreSQL e pgvector · OpenAI · rotinapet.vilquer.dev · github.com/caiovilquer/rotina-pet

- Construiu o backend de forma que cada família só alcança os próprios dados e cada perfil só o que lhe cabe, com regra de negócio isolada da infraestrutura: Kotlin e Spring Boot em arquitetura hexagonal com 7 módulos Gradle.
- Fez o assistente responder só com base nos documentos que aquele usuário pode ver, citando a evidência ou admitindo que não sabe: busca híbrida no PostgreSQL unindo full-text em português e busca vetorial, com permissão aplicada antes do ranking.
- Garantiu que nenhum documento se perca nem apareça pela metade na busca se o processamento falhar no meio: indexação em segundo plano com reprocessamento automático, fila de erros e troca atômica do conteúdo antigo.
- Permitiu que o tutor mostre o histórico clínico ao veterinário por um link que ele revoga quando quiser, sem o segredo vaziar em log de servidor nem ficar na barra de endereços: token viaja no fragmento da URL, é validado no formato, sai do endereço após a carga e anexos abrem por URL assinada de curta duração.
- Fez a evolução de peso e condição corporal ser lida também por quem não enxerga o gráfico: SVG próprio sem biblioteca de charts, com resumo em texto para leitor de tela e a mesma série disponível em tabela.
- Entrega planos de cuidado como rascunho que só vira dado real após confirmação do tutor, com resposta validada por schema e sem duplicata em reenvio; o CI roda testes de integração contra banco real e casos de prompt injection.

VIAZIO · Motor explicável de recomendação de viagens 03/2026 - 07/2026
Java 21 · Spring Boot · PostgreSQL · Resilience4j · React · viazio.vilquer.dev · github.com/caiovilquer/viazio

- Entrega uma recomendação que o usuário consegue questionar, com nota separada para data, destino e confiança do dado, e que redistribui o peso quando uma fonte externa está fora do ar em vez de fingir certeza.
- Manteve o ranking funcionando com uma fonte caída, isolando as cinco APIs externas para que a lentidão de um provedor não contamine as demais: cache, retry, circuit breaker e bulkhead por provedor.
- Fixou o teto de resposta em 2 segundos consultando cinco provedores: avalia candidatos em paralelo com virtual threads, responde com o que já está em banco e busca o que falta em segundo plano.
- Abriu a nota em suas partes na tela, para o usuário ver por que um destino ficou à frente do outro e o que faltou quando uma fonte não respondeu: interface em React alimentada pelo mesmo detalhamento que o motor calcula.
- Deixou a suíte de mais de 250 testes rodando em qualquer máquina e no CI sem depender de rede nem de chave de API, com deploy reproduzível por imagem Docker, migrações Flyway e métricas expostas via Prometheus.

FORMAÇÃO COMPLEMENTAR

Turing USP · Turing Academy, trilha de formação em IA do grupo de extensão da USP 03/2026 - 07/2026