Даниил Вдовин

Backend / Fullstack разработчик

+7 (982) 611-98-00 daniilvdovin4@gmail.com Telegram: @v_d_v_7 daniilvdovin.com career.habr.com/vdv007
github.com/VDV001 Екатеринбург от 150 000 ₽

ОБО МНЕ

Backend/Fullstack-разработчик с уклоном в Go и TypeScript. Полтора+ года коммерческой практики, параллельно веду несколько production-проектов на Clean Architecture + DDD + TDD - от AI-CRM и оценки проектов до модулей ERP. Пишу код, который понимают другие: четкие границы модулей, тесты, документация. Читаю чужой код и задаю вопросы до реализации.

AI - часть моего рабочего процесса каждый день (Claude Code CLI + настроенные MCP-серверы): планирование архитектуры, TDD, параллельная верификация через субагентов, кастомные скилы для команды. Расту в сторону сложных backend-систем и лид-роли: распределенные архитектуры, надежность, производительность, наставничество.

Екатеринбург 1.5+ года коммерческой разработки Русский - Родной
Английский - B1–B2, технический, активно развиваю УГГУ, Специалист по большим данным (Computer Vision)

ТЕХНОЛОГИИ

Backend	Go · Python · FastAPI · SQLAlchemy 2 · Alembic · Pydantic · Node.js · NestJS · chi · Gin · gRPC · PostgreSQL · Redis · pgx/v5 · golang-migrate · REST API · WebSocket · SSE · SMTP / IMAP · JWT / OAuth2 · bcrypt · CQRS · Event Sourcing · DDD · Clean Architecture · Repository / Unit of Work · TDD
Frontend	TypeScript · JavaScript · React 19 · Vite · Next.js 16 · App Router · Server Components · Zustand · SWR · TanStack Query · React Hook Form · Zod · Tailwind CSS · shadcn/ui · Three.js · PWA · i18n
DevOps & Инфра	Docker · Docker Compose · Docker Swarm · Kubernetes · Helm · kustomize · kind · Terraform / OpenTofu · tflint · GitHub Actions · CI/CD · Nginx · CloudFlare · Prometheus · Grafana · Loki · OpenTelemetry · MinIO / S3 · LocalStack · golangci-lint · Dependabot · CodeQL · govulncheck · gitleaks · SBOM · supply-chain gates · GoReleaser
Тестирование	testify · gomock · pytest · Vitest · Jest · Playwright · mutation testing · k6 · httptest · race detector · table-driven tests · negative control · fault injection · self-testing gates · проверка на копии живых данных
AI & Инструменты	Claude Code · Codex · MCP Servers · FastMCP · Claude Code Skills · Anthropic SDK · OpenAI · Groq · Gemini · Ollama · Prompt Engineering · RAG · pgvector · embeddings · semantic search · Subagent-Driven Development · prompt-injection protection · agent reliability gates · AI-driven TDD

ОПЫТ РАБОТЫ

Мультимедиа Видеосистемы - Fullstack-разработчик

Октябрь 2025 - настоящее время

- **Ведущий разработчик продукта заказчика**: около 2/3 коммитов за три месяца при восьми участниках репозитория
- Сквозной путь пресейла **бриф** → **ТЗ** → **спецификация** → **КП**: версии документов и публичная ссылка для согласования клиентом
- **Нормализованный каталог оборудования** с MCP-сервером на FastMCP: позиция публикуется только с заполненными ключевыми ТТХ, у каталога есть метрика готовности
- Типовые решения и **мастер подбора оборудования** по помещениям и подсистемам
- Экспорт КП в **1С (CommerceML 2.05)**
- Claude Code (CLI) как основной AI-инструмент: MCP-серверы (Serena, Context7, Playwright, Figma), **AI-driven TDD**, кастомные скилы для debugging и code review

Методологии разработки. Каждая задача T-xxx идет со спецификацией и тестами по CJM-сценариям. Архитектурные решения записываются в ADR.

Архитектура и проектирование. Backend на FastAPI со слоями api / models / schemas / repositories, миграции Alembic. Отдельный MCP-сервер каталога на FastMCP.

Backend (Python / FastAPI). SQLAlchemy 2, Pydantic 2, JWT + Argon2id, мультипроцесс через gunicorn и UvicornWorker. КП в трех вариантах: Base, Optimal, Premium.

Frontend (React / TypeScript). React 19, Vite 7, TanStack Query, Zustand, React Hook Form + Zod, Tailwind 4. Мастер подбора оборудования и окно типовых решений.

AI-инструменты. Бриф превращается в ТЗ через Codex, результат проверяет человек. Каталог наполняется через Claude Code и MCP-сервер.

DevOps и инфраструктура. CI в GitHub Actions, S3-совместимое хранилище (MinIO), PostgreSQL 16.

Тестирование. pytest, Vitest + RTL, Playwright e2e по CJM-сценариям, мутационное тестирование.

Интеграции. Экспорт КП в 1С (CommerceML 2.05), MinIO для файлов, публичные ссылки на ТЗ и КП для клиента.

Анвиртехно - Fullstack-разработчик

Ноябрь 2024 - настоящее время

- **Год+ стабильной удаленной работы** в команде: регулярный delivery, **code review с первого-второго раза**
- Автогенерация TypeScript-клиентов из OpenAPI: **устранен целый класс ошибок** рассинхронизации frontend/backend
- Реализовал PWA с offline-режимом и push-уведомлениями - **приложение работает без интернета и устанавливается как нативное**
- Подключил LLM-провайдеры (OpenAI, Ollama) в production с **fallback-стратегией и мониторингом токенов**
- Построил систему автобэкапов PostgreSQL + MinIO с **шифрованием и offsite-синхронизацией**
- Claude Code (CLI) как основной AI-инструмент: MCP-серверы (Serena, Context7, Playwright, Figma), **AI-driven TDD**, кастомные скилы для debugging и code review

Методологии разработки. API-first подход: OpenAPI спецификация → автогенерация TypeScript-клиентов. TDD для бизнес-логики. RDD для планирования фич перед реализацией. TypeScript-типы из схем БД.

Архитектура и проектирование. Clean Architecture на базе NestJS-модулей. REST API с автодокументацией из декораторов. Event-driven подход для слабосвязанных компонентов. Структура, готовая к выносу в микросервисы.

Backend (NestJS / TypeScript). NestJS как основной стек. JWT-аутентификация с refresh-токенами. Background jobs и scheduled tasks. LLM-провайдеры (OpenAI, Ollama) с fallback-стратегией и мониторингом токенов. Telegram Bot интеграция.

Frontend (React / TypeScript). Next.js PWA с offline-режимом и push-уведомлениями. Zustand + SWR для state management. React Hook Form + Zod для валидации. Компонентная библиотека в Storybook. i18n и темизация.

AI-инструменты. Claude Code (CLI) для code review, debugging и рефакторинга. MCP-серверы: Context7 (документация), Playwright MCP (E2E), Figma MCP (дизайн). TypeScript language server для code intelligence. Кастомные скилы.

DevOps и инфраструктура. Docker + Docker Compose, multi-stage builds. Автобэкапы PostgreSQL + MinIO с шифрованием и offsite-синхронизацией. CI/CD в GitHub Actions, Dependabot. Observability: Prometheus, Grafana, Loki.

Тестирование. Jest unit-тесты для TypeScript/Node.js. E2E на Playwright. Mocking для LLM-провайдеров и внешних сервисов. Тесты интегрированы в CI/CD: PR не мерджится без зеленой матрицы.

Интеграции. LLM-провайдеры (OpenAI, Ollama) с fallback и мониторингом. Telegram Bot API. S3/MinIO для хранения и бэкапов. Composio Gmail API для уведомлений.

ПРОЕКТЫ

Deal Sense - AI для тендерных команд

5725 коммитов, из них мои 5636 (замер 30.10) · v0.334.0 · 5 AI-провайдеров

Продуктовый AI-инструмент для оценки тендеров: скоринг go/no-go, генерация коммерческих предложений и разбор тендерной документации. Go 1.26 backend + React/Vite/TypeScript. Clean Architecture, DDD, pluggable LLM с 5 провайдерами и runtime-переключением.

Clean Architecture + DDD на Go 1.26: доменные инварианты в domain, порты в usecase, адаптеры в infra. Pluggable-провайдер LLM как доменный интерфейс. Модульная структура, готовая к расширению. golangci-lint v2 с анализатором modernize в CI

5 LLM-провайдеров (Anthropic Claude / OpenAI / Gemini / Groq / Ollama) с переключением в рантайме из конфигурации. Скоринг тендеров по рубрике (go/no-go), генерация КП из шаблонов, обработка тендерной документации, двухэтапная защита промпта от инъекций

Docker Compose, GitHub Actions CI/CD (build + race-тесты + линтер), multi-stage сборка. Локальная разработка в OrbStack. Версионирование через ldflags

Table-driven тесты, go test -race. Покрытие доменной логики, парсинга ответов LLM, скоринга и usecase-слоя. Тесты - гейт на merge в CI

ИС Академического Секретаря/Методиста

1640 коммитов · v0.223.0 · в production

Модульный монолит на Go: DDD + Clean Architecture, каждый домен инкапсулирован. In-memory EventBus с абстракцией под Kafka. CQRS + Event Sourcing + Repository + Unit of Work. 22 SQL-миграции PostgreSQL 17. Результаты: Redis-кэш горячих чтений через Decorator Pattern, мониторинг с нуля, автобэкапы с шифрованием, 189 релизов за 4 месяца.

Модульный монолит на Go: DDD + Clean Architecture, CQRS, Event Sourcing, Repository + Unit of Work. In-memory EventBus с абстракцией под Kafka. gRPC-интерфейсы. 22 SQL-миграции PostgreSQL 17 с версионированием и rollback

Gin: middleware-цепочки, correlation IDs. OAuth 2.0 + JWT (полные security claims, timing-attack prevention, bcrypt cost 14, rate limiting 5 req/15 min, 6 security headers). WebSocket чаты. Предиктивная аналитика. Composio Gmail API + Telegram Bot. 1C ERP-интеграция. Мультиязычность ru/en/fr/ar с RTL

Prometheus + Grafana (latency/error rate/throughput). Loki + Promtail (JSON логи). Grafana Tempo + OpenTelemetry (distributed tracing). Telegram-алерты: CPU/RAM/Disk/API errors/backup. Docker + Docker Swarm. GitHub Actions CI/CD. Nginx + CloudFlare. Автобэкапы PostgreSQL и MinIO с шифрованием

testify unit-тесты (Go), k6 load-тесты, Playwright E2E. AI-driven TDD: тесты до реализации, параллельная верификация. Документация: архитектура, API reference, deployment, security. GitHub Projects: 40+ Issues. .taskmaster - AI-assisted таск-менеджмент

Floq - AI Sales Assistant

731 коммит (замер 30.10) · v0.118.0 · 4 AI-провайдера

Full-stack B2B CRM с AI-генерацией холодных писем, мультиканальными секвенциями (Email + Telegram + звонки) и автоматической квалификацией лидов. Clean Architecture, DDD, 20 миграций PostgreSQL, 51 API-эндпоинт.

Clean Architecture (handler → usecase → repository → domain), DDD с value objects (LeadStatus, Channel, OutboundStatus), port/adapter interfaces, compile-time interface checks, TxManager для транзакций, DTO-маппинг без json-тегов в domain, 20 SQL-миграций (golang-migrate), анти-коррупционные слои (ProspectReader, LeadCreator)

51 REST API-эндпоинт на Go 1.26 + chi v5, JWT-авторизация (access + refresh), pluggable AI-провайдер (Claude/OpenAI/Groq/Ollama) с runtime-переключением из БД, SMTP-отправка через mail.ru/Yandex/Gmail (TLS 465 + STARTTLS 587), IMAP-polling входящей почты (go-imap/v2), Telegram MTProto клиент (gotd/td) для рассылки через личный аккаунт, email-верификатор (MX/SMTP probe/disposable filter), парсинг 2GIS API + website scraper, AI-чат ассистент с контекстом из БД, prompt feedback loop, tracking pixel для open/reply трекинга, cron-система фоллоуапов

Docker Compose (PostgreSQL 18 + Redis 8 + Ollama + Backend), GitHub Actions CI/CD (Go build/test + Next.js lint/build), multi-stage Alpine image, OrbStack для локальной разработки

130+ unit-тестов в 16 пакетах (testify, race detector), покрытие: auth (JWT middleware, token generation), AI client (extractJSON, prompt resolution, qualification parsing), email verification, settings DTO mapping, chat system prompt building

EstimatePro

414 коммитов · v0.15.0 · 145 тестов

Коллаборативная платформа для оценки проектов. Modular Monolith + Clean Architecture + DDD на Go 1.26 и Next.js 16 с PostgreSQL 18, Redis 8, MinIO. Telegram-бот Эсти с LLM-парсингом и двухэтапной архитектурой защиты промпта.

Modular Monolith с 7 модулями (auth, project, document, estimation, notify, bot, ws). Clean Architecture: domain → usecase → handler. DDD: entities с бизнес-логикой, sentinel errors, value objects. Адаптеры в composition root, zero cross-module imports. 8 PostgreSQL миграций через goose

57 REST endpoints (Chi router). JWT auth с token rotation в Redis. OAuth2 Google/GitHub. PERT-агрегация оценок. WebSocket real-time. Telegram-бот Эсти: 4 LLM-провайдера (Claude/OpenAI/Grok/Ollama), 12 intents, inline-кнопки, память разговоров, двухэтапная архитектура (classifier + personality formatter). Уведомления: in-app + email + Telegram

Docker Compose (PostgreSQL 18, Redis 8, MinIO, Go backend). GitHub Actions CI/CD. CORS allowlist, rate limiting (10 req/min на auth), MaxBodySize middleware, RequireProjectMember middleware для IDOR-защиты

145 тестовых функций в 25 файлах. Table-driven тесты, function-field mocks. Покрытие: все usecases, domain logic, handlers, LLM providers, Telegram client (httptest), webhook, session flows. TDD для estimation (parser + aggregation)

Lexis

148 коммитов · v0.17.0 · 4 AI-провайдера

AI-репетитор иностранных языков: модульный монолит на Go (Clean Architecture, DDD) + Next.js 16 фронтенд. 4 AI-провайдера, SM-2 интервальное повторение, JWT auth с token rotation, PostgreSQL, Redis.

Модульный монолит (4 модуля: auth, tutor, vocabulary, progress), слои domain → usecase → handler → infra. ProviderRegistry как доменный интерфейс. Автомиграции через golang-migrate с embedded SQL. Event bus (pub/sub) для межмодульной коммуникации. RFC 7807 Problem Detail для ошибок

JWT access/refresh token rotation с reuse detection и user-level blacklist через Redis. 4 AI-провайдера (Claude, GPT-4o, Qwen, Gemini) с SSE-стримингом. SM-2 spaced repetition для словаря. 5 режимов упражнений (chat, quiz, translate, gap, scramble). Валидация на уровне домена с sentinel errors

Docker Compose (PostgreSQL 18, Redis 8). GitHub Actions CI: golangci-lint, go test -race, Next.js build, Playwright e2e. Ldflags version injection. MaxBodySize, RequireJSON, RateLimit (prefix-based, fail-closed для login) middleware

83 unit-теста в 12 пакетах. Gomock для auth service, hand-written mocks для handler tests. Покрытие: token rotation, SM-2 алгоритм, goal progress, handler CRUD, eventbus concurrency, middleware auth. Playwright e2e для auth flow

kb-engine - движок базы знаний (OSS)

392 коммита, 58 тыс. строк Go, 246 тестовых файлов (замер 30.10) · 28 выпусков, 12 джоб CI

Открытый движок на Go: CLI, терминальный интерфейс и веб-витрина на React через go:embed - три поверхности поверх одного usecase-слоя, поэтому разойтись они не могут. 28 выпусков с бинарями под шесть платформ и SBOM. Развертывание в Kubernetes (kustomize + чарт Helm) и инфраструктура в Terraform. Лицензия AGPL-3.0-or-later с коммерческой альтернативой.

Clean Architecture + DDD: домен без часов и случайности, интерфейсы репозиториев объявлены в usecase-слое. Одно правило живет в одном месте - терминал и веб читают через общий query-слой, поэтому витрины не расходятся. Две карты архитектуры со ссылками file:line, которые проверяет валидатор в CI: движок - 84 узла и 48 сценариев, облачный слой - 35 узлов и 8 сценариев

Каталог записей с аудитами целостности, проверка живости ссылок, двусторонняя синхронизация jsonl ↔ xlsx и защитой от повторной записи прямо на пути записи. Поиск в четыре слоя: точное совпадение, транслитерация, расстояние редактирования и смысловой слой на векторных представлениях (индекс 1487 записей, модель bge-m3, служба вызывается по HTTP). Решение принято замером: смысловой слой закрывает один запрос из двенадцати, поэтому модель в бинарь не вносилась - четыре прямые зависимости и размер образа 13,5 МБ остались прежними

Kubernetes: kustomize (base + оверлеи) и чарт Helm, оба ставятся в один кластер kind в CI и сверяются по объектам в кластере. Проба живости и проба готовности разведены намеренно: испорченный каталог выводит под из Service, но не перезапускает его. Terraform: бакет S3 с версионированием и шифрованием, роль IAM только на чтение, состояние в бакете с блокировкой (проверено подсадкой чужого lock), tflint в CI. Эндпоинт /metrics в формате Prometheus без клиентской библиотеки, Prometheus и Grafana с дашбордом как кодом. GitHub Actions, 12 джоб: тесты с race-детектором, порог покрытия 80%, golangci-lint с modernize, govulncheck, CodeQL, gitleaks по всей истории, карантин на свежие зависимости с самопроверкой на настоящем инциденте. Проверено на LocalStack и MinIO; настоящий AWS и поведение под нагрузкой не проверялись

189 тестовых файлов, порог покрытия в CI. Каждый гейт проверен на подсаженном нарушении