

FRANCISCO ANTÔNIO DOS SANTOS GONÇALVES

DOS CARTÕES PERFURADOS À INTELIGÊNCIA ARTIFICIAL

50 ANOS DE UMA VIDA DENTRO
DA REVOLUÇÃO INFORMÁTICA

1975 - 2026

*"O código soa na minha mente
como música e sai pela ponta
dos meus dedos como se a minha
mente já o tivesse escrito."*

1975

Cartões Perfurados
COBOL • Assembler
IBM System/360

1980

ICL 1501
CDE/BTL
Software de Sistemas

1990

Unix • C/C++ • Redes
Telecomunicações
X.25 • ISDN • TCP/IP

2000

Linux • Web • Bases de Dados
Virtualização • Sistemas Distribuídos

2020

Python • Visão Computacional
Inteligência Artificial
Soluções Inovadoras

ASSEMBLER

COBOL

PASCAL

C / C++

Unix

Java

Python

Linux

Virtualização

Redes e Telecomunicações

Bases de Dados

Inteligência Artificial

Visão Computacional

Sistemas Distribuídos

IDEIAS
CÓDIGO
SOLUÇÕES
SEMPRE...

DOS CARTÕES PERFURADOS À INTELIGÊNCIA ARTIFICIAL

50 anos de uma vida dentro da revolução informática

Francisco António dos Santos Gonçalves

1975-2026

Prólogo - O código como música

Há pessoas que descobrem uma profissão. Outras descobrem, muito cedo, uma linguagem através da qual parecem compreender melhor o mundo. Comigo aconteceu a segunda coisa. Quando comecei a estudar informática e programação, aos dezassete anos, tive a sensação de que tinha encontrado algo que já me pertencia antes de eu saber dar-lhe um nome.

Programar nunca foi, para mim, apenas escrever instruções para uma máquina. É uma forma de organizar o pensamento. Primeiro surge o problema; depois as relações entre as partes; a seguir, quase sem dar por isso, começa a formar-se uma estrutura. Quando essa estrutura se torna suficientemente clara, o código aparece. Muitas vezes tenho a impressão de que já está escrito algures na mente e que os dedos se limitam a passá-lo para o teclado.

“O código soa na minha mente como música e sai pela ponta dos meus dedos como se a minha mente já o tivesse escrito.”

Durante meio século, essa música mudou de instrumentos. Começou nos cartões perfurados e no processamento batch. Passou pelo assembler de processadores proprietários, pelo COBOL e pelo Pascal, pelos terminais inteligentes, pelas fitas magnéticas e pelos mainframes. Depois vieram Unix, C, C++, Shell, bases de dados, redes X.25, ISDN e TCP/IP. Mais tarde chegaram Linux, virtualização, Web, Python, visão computacional, aprendizagem automática, containers e inteligência artificial.

Este livro não pretende ser uma história geral da informática. É a história de uma vida profissional passada por dentro dessa transformação. É também uma tentativa de registar uma época em

que programar exigia conhecer intimamente a máquina e em que cada byte tinha peso; e de a ligar ao presente, no qual um computador pode reconhecer rostos, interpretar imagens, conversar em linguagem natural e executar modelos de inteligência artificial localmente.

Ao olhar para trás, vejo tecnologias que nasceram, dominaram uma geração e desapareceram. Vejo empresas, sistemas operativos, protocolos e linguagens. Mas vejo sobretudo uma continuidade: a curiosidade. Foi ela que me levou aos computadores em 1975 e é ela que, cinquenta anos depois, continua a fazer-me experimentar, construir e perguntar: “E se tentarmos fazer isto de outra maneira?”

1. 1975 - Quando um programa cabia num maço de cartões

A minha história profissional em informática começa em 1975. Nesse ano iniciei a formação na Informax - Instituto Português de Informática, na Rua Castilho, em Lisboa. Era um mundo radicalmente diferente daquele que hoje associamos à palavra computador. Não havia um computador pessoal em cima da secretária, nem um editor interativo onde se pudesse alterar uma linha e executar novamente o programa alguns segundos depois.

Aprendíamos COBOL e IBM Assembler. O programa era material: existia num conjunto de cartões perfurados de 80 colunas. Era necessário pensar antes de executar. Os cartões eram preparados, organizados e enviados em lote para processamento num IBM System/360 instalado nos centros de dados da CP. Um erro podia significar um abend e a necessidade de corrigir o cartão e esperar pelo ciclo seguinte. A programação ensinava, por necessidade, disciplina mental.

Tive como professores e mentores Hélder Veríssimo e António Conceição. A exigência era elevada e o contacto com a programação produziu em mim um efeito imediato: senti que tinha encontrado a área em que queria trabalhar. Terminei a formação com uma classificação próxima da excelência - 97% no registo desta história profissional - e, antes mesmo do encerramento formal do curso, fui convidado pela direção da Informax a integrar a empresa.

Esse convite marcou a passagem entre aprender informática e viver dentro dela. Eu ainda era muito novo, mas a informática também o era. Em Portugal, as organizações que possuíam capacidade computacional relevante eram poucas; os recursos eram caros e centralizados; e quem programava tinha de compreender simultaneamente lógica, dados, limitações físicas e funcionamento operacional.

A disciplina imposta pela escassez

Hoje é fácil esquecer o efeito formativo das limitações. Memória, armazenamento e tempo de máquina eram recursos preciosos. Não se podia compensar um algoritmo pouco eficiente comprando simplesmente mais RAM ou mais CPU. Era necessário saber o que a máquina fazia, quanto custava cada operação e como os dados circulavam.

Essa escola marcou toda a minha carreira. Décadas mais tarde, quando trabalhei com servidores, virtualização, redes, bases de dados ou inteligência artificial, continuei a olhar para os sistemas como conjuntos de recursos finitos que devem ser compreendidos e utilizados com critério.

2. 1978 - Da programação à arquitetura dos sistemas

A partir de 1978, a minha atividade na Informax deixou de estar limitada à escrita de programas. A formação em análise funcional e análise orgânica obrigou-me a subir um nível de abstração. Já não bastava perguntar como escrever determinado algoritmo; era necessário compreender o processo de negócio, representar fluxos, definir ficheiros, formatos de registo, memória, interfaces e encadeamento operacional.

Foi também uma época de transição tecnológica. Os cartões perfurados começaram a ceder espaço às fitas magnéticas e a Informax expandiu a operação de recolha de dados para instalações na Avenida de Roma, em Lisboa. A informática começava a sair do modelo exclusivamente centralizado, embora os grandes centros de processamento continuassem a dominar.

Coordenei operações de recolha inteligente de dados com terminais ICL 1501. Estes equipamentos utilizavam armazenamento local em cassete ou fita magnética e eram programados numa linguagem de muito baixo nível, CDE/BTL, próxima do assembler. As limitações de memória eram severas. Cada rotina tinha de justificar o espaço que ocupava.

O algoritmo das raízes

Um dos problemas técnicos que mais recorde surgiu quando necessitávamos de calcular raízes quadradas e cúbicas para correlações estatísticas. O equipamento não possuía instruções nativas adequadas. A solução exigia construir o cálculo a partir das operações elementares disponíveis.

Uma revista Science & Vie descrevia o princípio de aproximação usado em calculadoras da Texas Instruments, baseado em somas sucessivas e rotações de bits. A ideia ficou-me na cabeça. Adaptei-a às possibilidades do CDE/BTL e desenvolvi uma rotina suficientemente rápida para ser utilizada nos terminais. Para mim, este episódio resume uma característica da engenharia informática daquela época: perante uma limitação do hardware, não havia

biblioteca a importar. Era preciso compreender o problema e inventar uma solução compatível com a máquina.

Confelis e as noites diante do System/360

Outro projeto importante foi o sistema de gestão de stocks para a empresa têxtil Confelis, desenvolvido em colaboração com o engenheiro Etelviro Alves Veiga, que tinha sido meu professor. O processamento utilizava tempo de máquina alugado num IBM System/360 da Data, empresa de processamento ligada ao setor financeiro.

Foram noites longas. O código COBOL era preparado em cartões e o trabalho prolongava-se diante da consola física do mainframe. Ali, entre comandos, memória, jobs e erros, o computador deixava de ser uma abstração. Era uma máquina concreta, cara, partilhada e exigente. A relação entre programador e sistema era direta.

3. 1980 - A ICL e a passagem para a engenharia internacional

Em 1980 surgiu um novo salto. A ICL Computers Portugal, cuja sede se encontrava na Avenida dos Estados Unidos da América, convidou-me para integrar os seus quadros. A mudança colocou-me em contacto com uma engenharia de sistemas de escala internacional e com centros tecnológicos no Reino Unido e nos Estados Unidos.

Passei por Utica, no estado de Nova Iorque, um centro com a herança tecnológica da Cogar Corporation, entretanto adquirida pela ICL, associado à evolução dos sistemas da família ICL 1500. No Reino Unido trabalhei em ambientes de engenharia de software de sistemas, nomeadamente em Feltham e, ao longo dos anos, em contacto com centros como Bracknell, Stevenage, Manchester, Londres, Cardiff e Beaumont College, em Old Windsor.

Uma das realizações que mais significado teve para mim foi o desenvolvimento de um Command Line Interpreter. O programa acabou integrado no sistema operativo da linha comercial ICL DRS 20. Para alguém que tinha começado poucos anos antes a entregar cartões perfurados para execução num computador distante, ver software próprio tornar-se parte de um sistema comercial representava uma mudança extraordinária.

Fazer computadores diferentes falarem entre si

Outra área decisiva foi a comunicação com o universo IBM. Trabalhei com emulação e protocolos compatíveis com os terminais IBM 2780 e 3780, utilizados em Remote Job Entry. A adaptação permitia comunicação programa-a-programa entre sistemas ICL distribuídos e mainframes IBM.

Este problema - fazer máquinas, arquiteturas e organizações diferentes comunicarem - acompanharia grande parte da minha vida profissional. Primeiro eram protocolos proprietários e ligações remotas; depois viriam X.25, redes bancárias, ISDN, TCP/IP e, finalmente, a Internet como infraestrutura universal.

4. Dos sistemas proprietários ao Unix

Durante os anos seguintes trabalhei em projetos para diferentes setores: banca, seguros, transportes, retalho, indústria e organismos públicos. COBOL acompanhou muitos desses sistemas, sobretudo em aplicações de gestão e processamento transacional. Pascal permitiu-me construir soluções de front-office e aplicações comerciais. Em níveis mais próximos do equipamento, o assembler e o desenvolvimento de drivers continuavam indispensáveis.

A diversidade dos projetos obrigava a ligar software a objetos físicos: terminais bancários, periféricos de balcão, leitores óticos,

balanças eletrônicas, sistemas de lotes e outros equipamentos. Foi uma aprendizagem importante. Um sistema de informação não termina no ecrã. Há sempre uma fronteira onde o software encontra o mundo real.

A chegada do Unix System V da AT&T foi, para mim, uma mudança conceptual. O sistema apresentava uma elegância diferente: ferramentas pequenas, composição por pipes, shell scripting, processos, ficheiros, permissões e uma cultura em que o programador podia construir rapidamente utilitários para automatizar o próprio ambiente. Trabalhei com Shell, C, C++ e Perl, e essa filosofia influenciou profundamente a forma como passei a pensar a engenharia de sistemas.

Redes antes da Internet

Antes de TCP/IP se tornar onnipresente, as comunicações de dados eram um território de protocolos e redes muito diferentes. Trabalhei com X.25 em arquiteturas de interligação, incluindo ambientes bancários. Na Segurança Social do Porto participei em soluções de comunicação entre delegações e pontos descentralizados através de bridges e linhas dedicadas de 64 kbps. Na Caixa Geral de Aposentações, a preocupação era encaminhar de forma controlada o tráfego de delegações do Norte para os sistemas centrais em Lisboa.

Mais tarde, ISDN e finalmente TCP/IP alterariam a escala e a simplicidade aparente da conectividade. Para quem tinha conhecido comunicações síncronas e assíncronas, protocolos ICL e IBM e redes X.25, a Internet não surgiu como magia. Surgiu como mais uma etapa - enorme - de um problema antigo: transportar informação de forma fiável entre sistemas.

5. Bases de dados, telecomunicações e sistemas distribuídos

A evolução dos sistemas de informação trouxe as bases de dados relacionais para o centro da arquitetura. Tive formação na Holanda em ICL Ingres e acompanhei a passagem de uma informática dominada por ficheiros e processamento batch para aplicações em que os dados passaram a ser partilhados, pesquisados e atualizados de forma cada vez mais dinâmica.

Na década de 1990 e no início do novo século, a minha atividade aproximou-se também das telecomunicações. Na CodiSoft e posteriormente na Isiware trabalhei em software, arquitetura de sistemas e soluções que cruzavam informática e comunicações. Foram anos de tecnologias como Voice over Frame Relay e Voice over IP, de integração entre plataformas e de utilização de linguagens e ferramentas como Visual Basic, C++, Oracle e SQL Server.

Linux começou entretanto a assumir um papel cada vez mais importante. A familiaridade anterior com Unix tornou essa transição natural. O que antes dependia de sistemas proprietários dispendiosos podia agora ser construído sobre plataformas abertas, hardware mais acessível e uma comunidade global de desenvolvimento.

6. 2001-2010 - Da aplicação à infraestrutura

No início dos anos 2000 continuei a trabalhar em ambientes Unix e COBOL, incluindo integração com equipamentos e processos físicos. Mas a escala dos sistemas estava a mudar rapidamente. A

infraestrutura passava a ser, ela própria, software.

No BANIF, entre 2002 e 2010, a minha atividade centrou-se fortemente na infraestrutura de tecnologias de informação. A virtualização permitiu enfrentar um problema que se tornara comum nos centros de dados: centenas de servidores físicos, cada um com utilização parcial, custos de energia, manutenção, espaço e administração.

Particpei numa consolidação que levou mais de duzentos servidores Intel para uma arquitetura muito mais concentrada, suportada por oito servidores IBM em alta disponibilidade. A virtualização - com tecnologias que ao longo da minha experiência incluíram VMware e, mais tarde, KVM e Proxmox - mudou a unidade fundamental do centro de dados. O servidor deixou de ser necessariamente uma caixa. Passou a ser uma entidade lógica que podia ser criada, movida, copiada, protegida e recuperada.

Para alguém que tinha começado numa época em que o acesso a um mainframe era comprado em horas, havia uma ironia histórica fascinante: décadas depois, uma única infraestrutura física podia simular dezenas ou centenas de computadores independentes.

7. A Web torna-se plataforma

A Web começou como uma forma de publicar documentos e acabou por se transformar numa plataforma universal de aplicações. Essa transformação voltou a aproximar-me intensamente do desenvolvimento. HTML, JavaScript, Java, PHP e outras tecnologias permitiam construir sistemas que já não dependiam de um terminal específico ou de um cliente proprietário instalado em cada posto.

Em 2014 trabalhei numa plataforma Web de marketing, recorrendo a Java e JavaScript. Noutros projetos, estas tecnologias

foram aplicadas a gestão promocional, orçamentação e automação industrial. O navegador tornava-se a nova consola universal.

Apesar das sucessivas mudanças de linguagem, a essência permanecia familiar. Um programa continuava a ser uma transformação rigorosa de estados e dados. O que mudava era a escala: mais utilizadores, mais camadas, redes globais, interfaces gráficas, serviços e bases de dados distribuídas.

8. Python, visão computacional e máquinas que começam a ver

Python entrou na minha caixa de ferramentas com uma característica que sempre valorizei: permite passar rapidamente da ideia ao protótipo sem impedir que se construa software sério. Usei-o em segurança, tratamento de imagens, automação e geração de documentos, incluindo PDF e Excel.

Em 2019, no projeto SUREIA, o percurso iniciado décadas antes com dados codificados em cartões encontrou uma realidade quase inimaginável em 1975: computadores capazes de analisar imagens. Trabalhei com Python e JavaScript, PostgreSQL, TensorFlow, OpenCV, Keras e YOLOv3, incluindo execução em Raspberry Pi e mecanismos de reconhecimento facial com tempos de resposta inferiores a poucos segundos.

A visão computacional despertou-me particularmente porque fecha um círculo. Nos primeiros sistemas, o mundo tinha de ser cuidadosamente traduzido para códigos que a máquina pudesse ler. Com redes neuronais e modelos de deteção, a máquina começou a extrair padrões diretamente de imagens do mundo físico.

9. O laboratório Voltaire - voltar a construir por prazer

Depois de décadas em empresas e projetos de clientes, a tecnologia continuou presente no meu cotidiano. O servidor a que chamei Voltaire tornou-se uma espécie de laboratório pessoal: Linux, KVM/libvirt, máquinas virtuais, redes, serviços, segurança, containers e inteligência artificial local coexistem numa infraestrutura que posso observar, alterar e reconstruir.

Construí um gestor de virtualização baseado em NiceGUI e libvirt, com controlo do ciclo de vida das máquinas virtuais, clonagem, backups, gráficos de CPU, memória e tráfego, edição de parâmetros, consola noVNC e mecanismos de previsão. O objetivo nunca foi apenas substituir uma ferramenta existente. Foi compreender o sistema e moldá-lo à minha maneira.

O Voltaire Shield nasceu da mesma lógica aplicada à segurança: observar tráfego, reputação de endereços IP, estados de ligações, logs de serviços, carga do sistema e saúde dos discos, e transformar esses sinais em capacidade de reação. Décadas depois das primeiras rotinas de baixo nível, continuo a sentir a mesma satisfação quando um conjunto disperso de sinais começa a formar um sistema coerente.

Vigilant-Guard: da câmara ao significado

No Vigilant-Guard, a infraestrutura de videovigilância deixou de ser apenas um conjunto de câmaras. Através de ZoneMinder, sockets Unix, YOLOv8, MediaPipe Pose, reconhecimento facial, MySQL e notificações, passei a experimentar uma cadeia em que a imagem é filtrada, uma pessoa é detetada, a pose pode ser analisada, o rosto recortado e comparado e um evento relevante comunicado.

É difícil não comparar este cenário com os terminais ICL 1501 da juventude. Na altura lutávamos por memória suficiente para executar uma rotina matemática. Hoje posso executar, numa infraestrutura pessoal, algoritmos de visão computacional que processam milhões de parâmetros. A diferença de escala é quase absurda; a lógica de engenharia continua reconhecível.

Conversar com a máquina

Com o ChatAI-eLabs comecei a explorar outra fronteira: sistemas capazes de trabalhar com linguagem. Organizar conversas, indexá-las, pesquisar milhares de pares de perguntas e respostas, alternar entre componentes locais e serviços externos, medir custos e acrescentar voz em português europeu são problemas diferentes dos que enfrentei nos anos 1970, mas voltam a combinar aquilo que sempre me atraiu: dados, arquitetura, interfaces e experimentação.

Ao mesmo tempo, projetos como o Dockumo, para gestão de Docker, e um sistema próprio de notificações para Android mostram uma característica que nunca desapareceu: quando uma ferramenta me faz falta, a primeira pergunta continua a ser se a posso construir.

10. 2026 - A inteligência artificial chega ao computador pessoal

A inteligência artificial contemporânea representa uma mudança diferente das anteriores. Não é apenas uma máquina mais rápida ou uma rede com maior largura de banda. Pela primeira vez, utilizamos sistemas que lidam com linguagem, imagens e padrões complexos de uma forma que, para o utilizador, se aproxima de capacidades cognitivas.

Interessa-me especialmente a possibilidade de executar inteligência artificial localmente. Há nisso uma continuidade com a cultura dos sistemas abertos: possuir a infraestrutura, compreender o que nela acontece, controlar os dados e escolher os modelos. O crescimento do ecossistema open source permite que capacidades que há poucos anos pertenciam apenas a grandes laboratórios possam ser experimentadas numa workstation ou num servidor privado.

O Voltaire ainda tem limites, naturalmente. Mas o simples facto de discutir quanta memória, que GPU e que quantização são necessárias para executar um modelo de linguagem local mostra até onde chegámos. Em 1975, eu aguardava que um lote de cartões fosse processado num System/360 de outra organização. Cinquenta anos depois, posso considerar executar um modelo de inteligência artificial dentro da minha própria infraestrutura.

11. O que cinquenta anos de informática me ensinaram

A tecnologia muda; os fundamentos resistem. Linguagens, fabricantes e produtos desaparecem. Estruturas de dados, concorrência, latência, memória, estados, interfaces, segurança e clareza lógica continuam a reaparecer sob nomes diferentes.

A abstração é poderosa, mas não deve esconder a realidade. Quem conheceu a máquina por baixo do software aprende a desconfiar de abstrações perfeitas. Uma cloud continua a ser feita de computadores. Um container continua a executar processos. Uma inteligência artificial continua a consumir memória, processamento, energia e dados.

A interoperabilidade é uma história interminável. Dos protocolos ICL/IBM ao X.25, do ISDN ao TCP/IP, das APIs aos serviços

distribuídos, grande parte da informática consiste em permitir que sistemas construídos por pessoas diferentes consigam cooperar.

O open source mudou a distribuição do poder tecnológico. Unix ensinou-me a força de ferramentas simples e combináveis. Linux e o software aberto levaram essa filosofia a uma escala mundial. Hoje é possível montar laboratórios de virtualização, segurança, visão computacional e IA com tecnologias que podem ser estudadas e modificadas.

A curiosidade envelhece melhor do que qualquer tecnologia. A competência mais duradoura não é conhecer uma determinada linguagem. É continuar disposto a aprender a próxima, sem esquecer o que as anteriores ensinaram.

12. O programador que ficou

Ao longo da carreira fui programador, analista, arquiteto de sistemas, engenheiro de comunicações, responsável por infraestrutura e construtor de soluções. Trabalhei com máquinas que hoje pertencem à história e com tecnologias que ainda estão a nascer. Contudo, se tiver de escolher uma palavra que atravesse todas essas funções, continuo a escolher programador.

Não porque tenha passado cinquenta anos permanentemente a escrever código, mas porque programar moldou a forma como penso. Perante um problema, procuro estados, relações, exceções, dependências e caminhos possíveis. Tento decompor a complexidade até encontrar uma estrutura que possa ser compreendida e controlada.

Talvez seja por isso que nunca senti a informática apenas como profissão. Quando programo, a mente pode viajar livremente e, ao mesmo tempo, permanecer presa a um objetivo concreto. Há imaginação, mas há também uma prova implacável: no fim, o

sistema funciona ou não funciona.

Esse equilíbrio entre liberdade e rigor acompanhou-me desde os primeiros cartões perfurados. O computador mudou de forma centenas de vezes. A sensação de construir alguma coisa que antes não existia permanece praticamente intacta.

Epílogo - A mesma curiosidade

Cinquenta anos são tempo suficiente para assistir ao desaparecimento de quase todas as certezas tecnológicas de uma juventude. O mainframe deixou de ser o único centro possível. O terminal tornou-se computador pessoal; o computador pessoal tornou-se nó de uma rede mundial; os servidores tornaram-se máquinas virtuais; as aplicações migraram para a Web; o software aprendeu a reconhecer imagens e a gerar linguagem.

Tive o privilégio de atravessar essas mudanças não apenas como observador, mas como alguém que programou, integrou, instalou, depurou, arquitetou e reconstruiu sistemas em várias dessas eras.

Não sei qual será a próxima transformação. Sei apenas que, enquanto tiver curiosidade suficiente para perguntar como funciona e vontade suficiente para tentar construir uma versão minha, a viagem não terminou.

Em 1975, diante dos primeiros cartões perfurados, eu não podia imaginar o mundo tecnológico de 2026. Se pudesse falar com aquele jovem de dezassete anos, talvez não lhe explicasse os modelos de linguagem, a visão computacional ou a virtualização. Dir-lhe-ia apenas: continua. A música ainda vai tocar durante muito tempo.

Cronologia tecnológica pessoal

1975-1977	Formação na Informax, Lisboa: COBOL, IBM Assembler, cartões perfurados e processamento batch em IBM System/360.
1978	Análise funcional e orgânica; recolha de dados; terminais ICL 1501; CDE/BTL; fitas magnéticas; projeto Confelis.
1980 e anos seguintes	ICL Computers: software de sistemas, CLI, IBM 2780/3780, engenharia internacional e evolução das famílias DRS.
Décadas de 1980-1990	COBOL, Pascal, drivers, periféricos, banca, seguros, transportes, retalho; Unix System V, C/C++, Shell e Perl; X.25 e redes de dados.
1994-2001	CodiSoft e Isiware: software, arquitetura TI, telecomunicações, VoFR/VoIP, bases de dados e integração.
2001	SolSuni: COBOL/Unix e integração com equipamentos e processos operacionais.
2002-2010	BANIF: infraestrutura, alta disponibilidade e consolidação/virtualização de mais de 200 servidores Intel.
2014	Plataforma Web de marketing com Java e JavaScript.
2019	SUREIA: Python, PostgreSQL, TensorFlow, OpenCV, Keras, YOLOv3 e Raspberry Pi.
2024-2026	Voltaire, KVM/libvirt, NiceGUI, Voltaire Shield, Vigilant-Guard, Docker/Dockumo, notificações Android e experiências com IA local.

Primeira edição autobiográfica - 2026