



Memórias da Linha de Comando

Publicado em 2026-09-12 12:15:00



Fragmentos do Caos · Memória Tecnológica

Memórias da Linha de Comando

Quando uma geração de engenheiros ajudava a construir o futuro digital

Há épocas da nossa vida profissional cuja importância só compreendemos verdadeiramente muitos anos depois.

Enquanto as vivemos, não pensamos que estamos a participar na História. Pensamos no programa que não compila, numa rotina de Assembler que ocupa memória a

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

E pensamos naquela pergunta ancestral de qualquer programador:

"Porque raio isto não funciona?"

No início da década de 1980, entre os Estados Unidos e o Reino Unido, tive o privilégio de trabalhar num desses períodos extraordinários.

Uma parte importante dessa experiência aconteceu no desenvolvimento do **ICL DRS20**, entre **Utica**, no estado de Nova Iorque, e os centros de engenharia da ICL em **Feltham** e **Bracknell**, no Reino Unido.

Olhando hoje para trás, percebo que nós, programadores e engenheiros de software daquela geração, estávamos a construir algumas das primeiras peças de um mundo que hoje consideramos completamente normal.

Na altura, porém, estávamos apenas a tentar fazer as máquinas funcionar.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

A história começa na região de ~~Old~~ ~~Rome~~, no estado de Nova Iorque.

Ali existia uma fábrica com uma história tecnológica interessante, ligada sucessivamente à Cogar Corporation, à Singer Business Machines e depois à britânica **ICL, International Computers Limited**.

Era um mundo bastante diferente dos atuais centros de desenvolvimento tecnológico.

Não havia campus futuristas, paredes de vidro, mesas de matraquilhos ou departamentos dedicados a descobrir nomes novos para coisas que já existiam.

Havia placas eletrónicas. Osciloscópios. Terminais. Impressoras. Discos e disquetes. Diagramas de circuitos. Manuais técnicos. E enormes pilhas de listagens em papel contínuo.

Acima de tudo, havia **engenheiros**.

Engenheiros de hardware. Engenheiros de software. Programadores de sistemas. Especialistas em comunicações. Programadores de aplicações. Técnicos de teste.



utilizável.

Uma geração que programava perto da máquina

Desenvolver software naquela época significava conhecer intimamente o computador.

Não havia camadas intermináveis de abstração a separar o programador da eletrónica.

Quem trabalhava em sistemas precisava de compreender processadores, registos, endereçamento, interrupções, buffers, dispositivos e estruturas de memória.

O **Intel 8085** não era uma abstração escondida atrás de dezenas de frameworks. Estava praticamente diante de nós.

Cada rotina tinha consequências. Cada byte interessava. Cada acesso à memória tinha um custo. Cada estrutura precisava de justificar a sua existência.

E se algo corresse mal, a máquina não preparava um relatório de diagnóstico com sugestões amáveis.



Os computadores possuem esse desagradável hábito de interpretar literalmente as instruções humanas.

Construir um sistema operativo distribuído

O **DRS20** representava uma abordagem particularmente interessante à computação empresarial.

A ideia de distribuir recursos por várias estações ligadas em rede era, naquele princípio dos anos 80, muito mais ousada do que hoje conseguimos imaginar.

Atualmente vivemos dentro de sistemas distribuídos: cloud, microserviços, clusters, storage distribuído e aplicações que recorrem simultaneamente a serviços espalhados por diferentes centros de dados.

Mas nessa altura ainda estávamos a descobrir como ligar pequenas máquinas inteligentes e fazê-las colaborar.

Era necessário desenvolver gestão de dispositivos, sistemas de ficheiros, comunicações, protocolos de rede, interfaces de operação, linguagens, ferramentas de

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

O sistema operativo precisava de conversar com o hardware. As aplicações precisavam de conversar com o sistema operativo. As máquinas precisavam de conversar entre si.

E os utilizadores precisavam de conseguir conversar com tudo aquilo sem possuir um doutoramento em arquitetura de computadores.

Essa última parte parecia pequena. **Não era.**

Quando foi preciso ensinar o computador a falar com o operador

Um dos problemas que surgiu na passagem para o novo ambiente era simples de formular:

como deveria o utilizador comandar o sistema?

Era necessária uma interface de linha de comandos. Uma **CLI, Command Line Interface.**

Hoje parece óbvio que um sistema operativo possua comandos para copiar ficheiros, consultar dispositivos, executar programas ou manipular informação.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

dispositivo? Como referenciar um ficheiro? Como distinguir a origem do destino? Como tratar comandos inválidos? Como fazer tudo isso utilizando o mínimo possível de memória?

Foi aí que surgiu uma das componentes em que trabalhei pessoalmente: a **CLI do Native Mode do DRS20**.

E tive a felicidade profissional, hoje provavelmente considerada uma atividade ligeiramente subversiva, de poder avançar com bastante iniciativa técnica.

copy #1.file1 #2.file2

Uma instrução como

```
copy #1.file1 #2.file2
```

resume muito bem a filosofia daquela época.

O cardinal seguido de um número, como #1 ou #2, permitia identificar diretamente um dispositivo. Depois vinha o nome do ficheiro.

Simples. Curto. Determinístico. Poucos caracteres para interpretar. Pouca memória necessária. Pouca ambiguidade.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

A elegância estava precisamente na ausência de desperdício.

Quando uma instrução estava errada, também não havia motivo para escrever uma dissertação. A resposta podia ser simplesmente:

`invalid command`

Problema perfeitamente descrito. O comando era inválido. Corrigia-se. Continuava-se.

Nenhum psicólogo de interfaces teria aprovado semelhante brutalidade textual, mas a memória agradecia profundamente.

A engenharia do pouco

Existe algo que aquela geração aprendeu talvez melhor do que qualquer outra:

escassez obriga a pensar.

Quando existe memória praticamente ilimitada, é fácil resolver problemas acrescentando mais camadas.



Blogue Fragmentos do Caos

A verdade nasce onde o pensamento é livre.

diferença. Uma estrutura mais compacta fazia diferença. Algumas dezenas de bytes podiam fazer diferença.

Não porque estivéssemos apaixonados por minimalismo. Porque não havia alternativa.

Programar significava constantemente negociar com a máquina.

Eu quero esta funcionalidade.

A máquina responde:

não tens memória.

Então fazemos de outra maneira.

Foi uma escola extraordinária de engenharia.

Mas ninguém construía aquilo sozinho

Com o passar dos anos, às vezes a História tecnológica fica perigosamente simplificada.

Escolhe-se uma pessoa. Um nome. Uma fotografia. E conta-se a história como se grandes sistemas tivessem aparecido da cabeça de um único génio solitário.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

Uns trabalhavam no kernel e nos componentes internos do sistema. Outros nas comunicações. Outros nos controladores. Outros nas linguagens. Outros nas aplicações. Outros nos testes.

Outros passavam dias a descobrir por que motivo o trabalho produzido pelos anteriores não fazia exatamente aquilo que estava escrito na especificação.

Cada um conhecia profundamente a sua parcela. E depois era necessário juntar tudo.

É precisamente aí que começam os verdadeiros problemas interessantes da engenharia.

Atravessar o Atlântico

O desenvolvimento do DRS20 rapidamente ultrapassou os limites de Utica.

A ICL era uma empresa britânica e grande parte da engenharia estava naturalmente concentrada no Reino Unido.



Blogue Fragmentos do Caos

A verdade nasce onde o pensamento é livre.

De repente, aquele projeto transformava-se numa experiência transatlântica.

Equipas americanas. Equipas britânicas. Engenheiros com diferentes formações. Diferentes hábitos de desenvolvimento. Diferentes maneiras de olhar para problemas semelhantes.

Isso era uma riqueza.

E por vezes, naturalmente, uma excelente forma de produzir discussões técnicas suficientemente longas para competir com qualquer sessão parlamentar.

Mas no final havia código para entregar. E isso costuma ser um excelente mecanismo de aproximação diplomática.

Feltham: colocar COBOL dentro do novo mundo

Em **Feltham**, uma das áreas em que participei esteve relacionada com a implementação do **CIS COBOL da Micro Focus** no ambiente do DRS20.

É fácil rir hoje do COBOL.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

E continua vivo, precisamente para irritar sucessivas gerações que anunciam solenemente a sua morte.

Nos anos 80, porém, COBOL era absolutamente fundamental.

Grande parte do software empresarial era escrita nessa linguagem: banca, contabilidade, faturação, gestão de stocks, seguros e administração.

Um computador destinado ao mercado empresarial precisava de executar software COBOL. Não havia discussão.

Por isso, colocar aquele ambiente de desenvolvimento no DRS20 significava muito mais do que simplesmente adicionar outra linguagem.

Significava tornar possível migrar e desenvolver **aplicações comerciais reais**.

Era a ponte entre a engenharia do novo sistema e as necessidades económicas das empresas.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

Depois havia Bracknell.

Bracknell representava para mim uma outra dimensão do projeto.

Ali sentia-se claramente a escala da engenharia de software da ICL.

Um computador empresarial não era apenas o sistema operativo. Era um ecossistema.

Havia ferramentas de programação, aplicações, utilitários, comunicações, software de escritório, sistemas de ficheiros, drivers, compiladores, editores e gestão de dispositivos.

Tudo precisava de coexistir.

E os engenheiros responsáveis por cada componente precisavam de compreender aquilo que os outros estavam a fazer.

Um sistema informático é uma criatura curiosa. Pode ter vinte componentes excelentes. Se dois deles não comunicarem corretamente, o utilizador conclui simplesmente:

"Isto não funciona."

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

Por isso, uma parte enorme do trabalho de engenharia acontecia precisamente nas fronteiras entre componentes. Onde um módulo terminava. E outro começava.

Essas fronteiras continuam hoje a ser alguns dos lugares mais perigosos de qualquer arquitetura informática.

Os engenheiros invisíveis

Recordo aquela época também por uma razão que considero importante.

A maioria dos engenheiros que construiu aqueles sistemas nunca aparecerá em livros de História.

Alguns nomes talvez sobrevivam em documentos técnicos. Outros em antigas listagens. Outros em manuais. Muitos desaparecerão completamente do registo público.

Mas estiveram lá.

Engenheiros que escreveram drivers. Engenheiros que desenvolveram protocolos. Engenheiros que criaram sistemas de ficheiros. Engenheiros que implementaram compiladores.



Blogue Fragmentos do Caos

A verdade nasce onde o pensamento é livre.

É assim que a tecnologia normalmente nasce.

Não através de um momento cinematográfico de inspiração.

Mas através de milhares de pequenas decisões corretas tomadas por muitas pessoas.

Antes da Internet, já estávamos a construir redes

Uma das coisas mais fascinantes do DRS20 era precisamente a ideia de rede.

Várias estações. Recursos partilhados. Comunicações através de uma LAN. Processamento distribuído.

Hoje qualquer adolescente vive constantemente ligado a sistemas milhares de vezes mais complexos.

Mas naquela altura estávamos a trabalhar praticamente na pré-história dessa realidade.

Era preciso pensar como os recursos seriam identificados. Como os nós comunicariam. O que aconteceria quando uma estação desaparecesse. Como

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

memória que hoje seriam insuficientes para carregar o logótipo de muitas aplicações modernas.

E funcionava.

É talvez isso que ainda hoje me fascina.

A liberdade para resolver problemas

Havia também uma cultura profissional que recordo com saudade.

Quando aparecia um problema técnico, muitas vezes era dada ao engenheiro liberdade para encontrar a solução.

Naturalmente existiam hierarquias. Prazos. Reuniões. Documentação. Gestão.

Mas havia espaço para uma coisa fundamental:

iniciativa.

Se alguém identificava uma maneira melhor de fazer algo, podia experimentar. Construir um protótipo. Demonstrar. Medir. Convencer.

Era uma cultura muito ligada ao resultado.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

Se funcionava e resolvia o problema, tínhamos algo interessante.

Não havia inteligência artificial para perguntar

Quando penso hoje naquela época, existe uma diferença particularmente divertida.

Se tínhamos uma dúvida, não podíamos perguntar a uma IA.

Não existia Google. Não existia Stack Overflow. Não havia fóruns globais com respostas para praticamente qualquer erro imaginável.

Havia **manuais**. Colegas. Documentação interna. Listagens. Experiência. E muitas horas de investigação.

Talvez por isso aprendêssemos determinadas coisas profundamente.

Quando alguém passa seis horas a descobrir por que razão um bit está errado num registo, tende a recordar durante bastante tempo para que serve aquele bit.



Mas eficiente.

Não sabíamos que era uma revolução

Hoje chamamos àquele período **revolução digital**.

Na altura ninguém entrava no escritório de manhã dizendo:

"Hoje vou trabalhar um pouco na revolução digital."

íamos trabalhar.

Havia versões para entregar. Bugs para resolver. Clientes à espera. Equipamento para testar. Programas para otimizar. E colegas de diferentes especialidades tentando manter tudo coerente.

Só muitos anos depois percebi que aquela soma de trabalhos aparentemente quotidianos estava a contribuir para uma mudança gigantesca.

O computador deixava progressivamente de ser apenas uma máquina centralizada reservada a especialistas.



Blogue Fragmentos do Caos

A verdade nasce onde o pensamento é livre.

Quarenta anos depois

Hoje posso ter sobre uma secretária uma máquina com capacidade computacional inimaginável naquela época.

Posso virtualizar dezenas de sistemas. Posso criar redes inteiras por software. Posso executar bancos de dados gigantescos. Posso falar com sistemas de inteligência artificial em linguagem natural.

E tudo isso parece normal.

Mas quando olho para a tecnologia atual, continuo a reconhecer princípios que aprendemos naquela época.

Compreender o problema antes de escrever código. Conhecer a arquitetura. Evitar complexidade desnecessária. Pensar nos recursos. Testar. Medir. Corrigir. Integrar.

E sobretudo compreender uma verdade que nenhuma revolução tecnológica tornou obsoleta:

software é trabalho coletivo.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

Seria uma afirmação absurda.

Antes de nós existiram gerações extraordinárias de matemáticos, engenheiros e cientistas. Depois de nós vieram outras gerações igualmente extraordinárias.

A tecnologia é uma enorme construção coletiva.

Mas tivemos a felicidade de trabalhar numa ponte particularmente interessante da História.

Entre o grande computador central e sistemas mais distribuídos.

Entre o terminal passivo e a estação inteligente.

Entre a máquina isolada e a rede.

Entre o especialista informático e o utilizador de escritório.

Entre a programação reservada a poucos e o mundo digital que acabaria por chegar a praticamente todos.

E ainda me lembro daquela linha

No meio de toda esta evolução tecnológica, uma coisa permaneceu curiosamente gravada na minha memória:

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

Alguns caracteres.

Mas nela consigo ver uma época inteira.

Uma época em que cada byte precisava de ser pensado.

Em que tínhamos recursos reduzidos e problemas enormes.

Em que dezenas de engenheiros espalhados entre **Utica, Feltham e Bracknell** iam construindo, componente após componente, uma nova forma de trabalhar com computadores.

Não havia cloud. Não havia Internet comercial. Não havia inteligência artificial. Não havia milhões de bibliotecas disponíveis à distância de um comando.

Havia conhecimento.

Havia curiosidade.

Havia engenheiros.

Havia liberdade para tentar.



Blogue Fragmentos do Caos

A verdade nasce onde o pensamento é livre.

chegar.

Provavelmente ninguém sabia.

Mas continuávamos a escrever código.

A testar.

A discutir arquiteturas.

A corrigir erros.

A melhorar sistemas.

Linha após linha.

Byte após byte.

Engenheiro após engenheiro.

**Até que, quase sem darmos por isso, o futuro
começou a arrancar.**

Francisco Gonçalves

Fragmentos do Caos · 28 de agosto de 2026

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

participação direta do autor no desenvolvimento da interface de linha de comandos do Native Mode do DRS20, à sintaxe recordada, ao trabalho realizado em Feltham e Bracknell e à dinâmica quotidiana das equipas são apresentadas como memória de quem viveu essa época, e não como reconstrução historiográfica independente.

As fontes abaixo documentam o contexto técnico e industrial mais amplo: a aquisição pela ICL dos ativos da Singer Business Machines e da operação de Utica; a evolução do DRS20 e do sistema operativo DRX; a utilização de processadores Intel 8080/8085 e de redes locais; a disponibilidade de CIS COBOL no ecossistema DRS20; a importância histórica de Bracknell para o desenvolvimento de software da ICL; e o papel duradouro do COBOL no software empresarial. Algumas memórias específicas do autor poderão não estar preservadas em documentação pública acessível, uma limitação normal na reconstrução da história quotidiana da engenharia de software das décadas de 1970 e 1980.

Referências internacionais e leitura complementar

1. [Computer Conservation Society — The Seizing of Singer, Michael Knight](#). História da aquisição da Singer

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

da linha Singer/System Ten para a estratégia DRS20 da ICL no início da década de 1980.

3. [Bitsavers / Computer History Museum — ICL Minicomputers: The Small Systems Story, Version 3.](#) Documento histórico detalhado sobre arquitetura DRS20, DRX, processadores Intel, networking, Native Mode, R-Mode e software, incluindo CIS COBOL.
4. [Computer History Museum — documentação histórica da Micro Focus.](#) Material de época que lista o ICL DRS20/DRX entre as plataformas suportadas pelo software COBOL da Micro Focus.
5. [Science Museum Group — International Computers Limited.](#) Enquadramento institucional da ICL, incluindo a presença histórica em Lovelace Road, Bracknell, e a evolução posterior da empresa.
6. [Computer Conservation Society — Computer Resurrection, Issue 21.](#) Testemunho histórico sobre a consolidação em Bracknell de uma das maiores instalações europeias de produção de software da época.
7. [Micro Focus — COBOL 60th Anniversary.](#) História e continuidade do COBOL enquanto linguagem de software empresarial e referência ao trabalho da Micro Focus desde 1976.
8. [Centre for Computing History — Martin Campbell-Kelly, ICL: A Business and Technical History.](#) Referência bibliográfica sobre a evolução empresarial

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.



Fragmentos do Caos:

[Blogue](#)

•

[Ebooks](#)

•

[Carrossel](#)

 Esta página foi visitada ... vezes.

[Contactos](#)