

Um pouco de História da Computação : Do ICL 1500 ao DRS 20: Bastidores de um Monitor de Comandos (Feltham, 1981)





um Monitor de comandos (Feltham, 1981)

Uma memória técnica pessoal sobre a transição de aplicações de gestão do ICL 1500 para o DRS 20 — e como um pequeno CLI em BASIC ajudou a atravessar a ponte.

Introdução

Em **1981**, em **Feltham (Reino Unido)**, fui destacado para validar, num ambiente **ICL 1500 com disco**, as aplicações que eu próprio tinha desenvolvido em Portugal — **Faturação, Stocks, Contas-correntes e Contabilidade POC** — com vista à sua migração para o **DRS 20**. O novo sistema arrancou em **modo "retained"**, emulando o interface do ICL 1500, e rapidamente senti falta de uma ferramenta mais direta para acelerar testes e operações. A resposta foi construir, em **BASIC comercial**, um **monitor de comandos** (CLI, command line interpreter & executor). Um ano depois, já com o DRS 20 em **modo nativo**, encontrei um monitor de comandos integrado cuja sintaxe era, para minha surpresa, **praticamente idêntica** à que tinha concebido.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

1) Contexto e objetivos

- **Local & data:** Feltham, UK — July 1981-1982
- **Missão:** validar e preparar a passagem de aplicações de gestão do **ICL 1500** para o **DRS 20**.
- **Aplicações em causa:** Faturação, Stocks, Contas-correntes, Contabilidade **POC**.
- **Meta imediata:** assegurar compatibilidade funcional e operacional, reduzir riscos de migração e ganhar velocidade nos ensaios.



Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

fricções: comandos menos diretos, ciclos de teste lentos e uma sensação de estar a trabalhar por camadas. Para validar com rigor (e sem desperdiçar tempo), era preciso uma ferramenta **scriptável, repetível e transparente**.

3) Porque um CLI?

Um **CLI** oferece três vantagens decisivas em contextos de migração e testes:

1. **Automação** — repetir cenários sem esforço, com logs claros.
2. **Precisão** — cada comando é explícito; fácil detetar diferenças entre execuções.
3. **Velocidade** — menos navegação por menus, mais foco na tarefa.

4) Construção em BASIC comercial

Escolhi **BASIC comercial** por estar disponível no ambiente e permitir resultados rápidos. A arquitetura foi deliberadamente simples:

- **Laço principal** a ler uma linha de comando e a despachar para uma **tabela de verbos**.
- **Parser enxuto**, baseado em tokens separados por espaços e vírgulas.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

inequívocas (sem códigos obscuros).

5) Gramática de comandos (exemplos)

A sintaxe seguia o padrão **VERBO argumentos**. Alguns exemplos da época:

```
copy #1,filenameinput #2.filenameout  
rename #1.testeabc #1.testeum  
list #2  
delete #2.filename  
run #1faturacao.cbl
```

Onde #x era o dispositivo de IO, diskete ou disco fixo.

Princípios de desenho:

- **Minimalismo** (o essencial primeiro, sem switches crípticos).
- **Previsibilidade** (mesma ordem de argumentos em todos os verbos que manipulavam ficheiros/dispositivos).
- **Feedback claro** (confirmações sucintas; erros com indicação do passo falhado).

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

depois **regressei a Portugal** e deixei o projeto para trás. Cerca de **um ano depois**, com o **lançamento do DRS 20 em modo nativo**, deparei-me com um **monitor de comandos integrado** no sistema cuja **sintaxe e espírito** eram uma **cópia fiel** do que eu implementara em Feltham. Foi uma surpresa, misto de orgulho e assombro.

Este relato não pretende julgar processos internos nem abrir discussões de propriedade intelectual; é apenas o registo honesto de uma sequência de factos vivida no terreno.



Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

- **Em migrações, simplicidade é uma arma:** comandos claros aceleram testes e reduzem ambiguidades.
- **Documentação é memória:** readmes, exemplos e logs são ouro passado um ano.
- **Portabilidade mental:** pensar em **interfaces estáveis** (ex.: #1, #2) ajuda a atravessar gerações de hardware.
- **Guardar provas de trabalho:** snippets, datas e versões facilitam futuras reconstruções históricas.

8) Linha temporal (aproximada)

- **1981** — Feltham: validação no DRS 20 em retained mode; desenvolvimento do CLI em BASIC; entrega do código a Terry Fuller.
- **1982** — Lançamento do **DRS 20 em modo nativo**; monitor de comandos integrado, com sintaxe praticamente idêntica.
- **Depois** — O capítulo encerra-se, mas as ideias (e a gramática) seguem viagem.

9) Glossário rápido

- **ICL 1500** — Sistema onde corriam as aplicações originais (com disco).
- **DRS 20** — Nova plataforma a lançar no mercado na época.
- **POC** — Plano Oficial de Contabilidade português, base das regras contabilísticas usadas.



10) Agradecimentos

À equipa de **Feltham** pelo ambiente de trabalho, e uma nota especial ao **Terry Fuller** pelo desafio técnico que, direta ou indiretamente, impulsionou este pequeno projeto.

11) Fontes e notas históricas

Abaixo ficam fontes públicas e testemunhos técnicos que ajudam a contextualizar o DRS 20, o DRX (Executive) e o arranque em “retained” vs “native” mode. A gramática completa do CLI do Executive não está facilmente disponível online; estes apontadores são um bom ponto de partida para investigação adicional.

Visão geral da gama DRS e do DRS 20

- ICL DRS — visão geral (modelos, CPUs 8085/80188/80186, arranjo de nós LAN coax 93 Ω a 1,25 Mbps, unidades com/sem disco e boot partilhado).
- Notas de evolução para a linha DRS (300/6000) — útil para não confundir com plataformas posteriores (DRS/NX Unix, etc.).

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

do 1500 em bit-slice) e o **"N-Mode"** (nativo).

- Testemunhos técnicos coevos (listas/foruns) confirmam a passagem de **retained** para **native (8085AH2)** no DRS 20 e apontam limitações de desempenho do retained.

Sistemas operativos e ferramentas

- **DRX — Distributed Resource Executive** como ambiente/"executive" do DRS 20; referências a edições como **DRX 2.1** e à disponibilidade de **CP/M 2.2** e ferramentas de desenvolvimento (Microsoft BASIC, CIS-COBOL, Pascal, Userbuild, Demon).

Arquivos e repositórios para aprofundar

- **Bitsavers** — repositório de manuais ICL (incl. sínteses técnicas e PDFs).
- **Science Museum Group Archives** — finding aid do fundo ICL, com séries de manuais e documentação comercial/técnica.
- **Museus e coleções online** (p.ex. Centre for Computing History) com entradas sobre a gama DRS.

Artigo [histórico] de **Francisco Gonçalves** in Fragmentos do Caos

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

