



Beyond the Search Bar: How AI Became My Co-Worker, In SAP and In Life

PRESENTED BY:
**Rogério Faleiros, Independent SAP Controlling
Consultant**

```
1 REPORT z_beyond_the_search_bar.  
2  
3 * twenty-one years of Controlling, and one question  
4 * that turned out to be the first line of something else  
5  
6 START-OF-SELECTION. ■
```

You have all written this line. Tonight it runs for ninety minutes.

WHO IS TELLING YOU THIS

Twenty-one years in Controlling. Six weeks as a **builder**.

THE CONSULTANT

Since 2005

Independent SAP Controlling consultant — food processing, chemical, construction, agribusiness

Since 2008

IFRS and CO-PC integration projects across the United States, Brazil and Germany

Certified

FI/CO consultant, MBA in IT

AND THE TWO BOOKS

SAP PRESS

Configuring Controlling in SAP ERP

SAP PRESS

Introducing the Material Ledger in SAP S/4HANA

Educator

Trains incoming SAP consultants in Brazil — which is why the course for teenagers later tonight is not a detour

Nothing in this talk was built by a developer. It was built by **someone who knows Controlling** and learned to ask well.

TONIGHT'S CO-AUTHOR

Three ways of working with the **same** colleague.

The conversation

Reasoning and judgment. The “let’s think this through together” layer — no files, nothing to install.

On the desktop

Real files and real tasks on a real machine. It built **this very presentation**.

In the repository

Plans, writes, tests and ships working software. It rebuilt a live Fiori report from scratch — you will see it tonight.

Tonight I am the narrator.

Claude did the work — you are about to watch it.

Same model underneath. What changes is how much of the work it is allowed to **touch**.

ASKED WHAT IT THOUGHT IT WAS FOR

“I’m not here to replace what makes you human.
I’m here to clear away **what doesn’t** —
so humanity can move to its next stage.”

— CLAUDE

TONIGHT

Five topics. One of them is **yours**.

1 How did it start?

Five thousand years of making the machine understand us

2 How did I start?

Every machine I ever had, and the one question that began it

3 Just a conversation

No files, no repository — and a local rival that lost

4 From files to finished work

The moment it got to touch the disk

5 Code with SAP

One transaction rewritten, and the commit that started everything

The last one is the SAP case. The other four are how it became possible.

TOPIC 1

How did it start?

From the abacus to the agent — a five-thousand-year run-up to tonight.

Five thousand years of making the machine **understand us.**



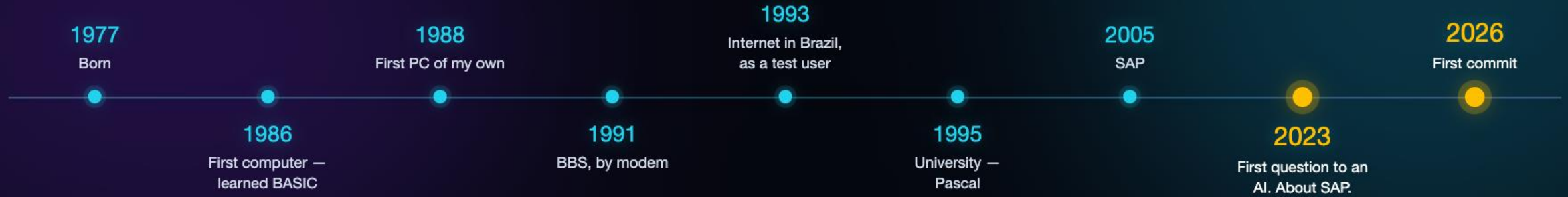
Every step shortened the distance between what a person meant and what a machine did. **This is only the latest one.**

TOPIC 2

How did I start?

A boy with a BASIC manual, and every machine since.

Every machine I ever had was somebody letting me **in**.



The last two are three years apart — and everything in this talk happens **between them**.

EVERY CONVERSATION I HAVE HAD WITH AN AI · 2023 - 2026

Three and a half years of **asking**. Nine weeks of **building**.



80,581

words typed · 1,630 chats · 3.5 years

916,003

words typed · 147 sessions · 63 days

x234

words per day

TOPIC 3

Claude — just a conversation

No files, no repository. Only what you can say out loud.

A MACHINE ON THE DESK · MAC MINI M4 PRO, 24 GB

I asked it to build me a rival. It told me the rival was **worse**.

WHAT ACTUALLY SHIPPED

Qwen3 14B · MLX 4-bit · 8.32 GB

Full GPU offload on the M4 Pro, 131k-token context, comfortable headroom

The load that failed

An 18.27 GB vision model was blocked outright — macOS keeps 4–6 GB for itself, and the model pushed past what was safe

Weighed, not guessed

Three candidates costed against the 24 GB ceiling before anything was downloaded

THE HONEST ANSWER

“The Claude you’re using right now is much better. The local model is roughly equivalent to Claude Haiku.”

Asked what the advantage of a local model even was — and answered against itself: privacy, offline, no usage limits. Not quality.

A tool that talks you out of the wrong tool — **including itself** — is not a search engine. That is a co-worker.

TOPIC 4

From files to finished work

The moment the conversation got to touch the disk.

THE FIRST THING WE EVER BUILT TOGETHER

Five hours. One **bank** at a time.

EVERY STATEMENT BROKE IT DIFFERENTLY

One bank

Missed refunds and incoming wires entirely

Another

Flipped the sign of its balance when the account went overdrawn

A third

Dropped instant-payment lines that carried no credit or debit letter

“It’s loading the balance-of-the-day line — that’s just information, not a transaction.”

A real bug report. Fixed the same day.

AND IT WENT LIVE THE BORING WAY

Local

One HTML file. Double-click, it runs. Every fix tested here first

Database

The same data whether opened from disk or from the live site

GitHub

The moment version control stopped being a term and became a habit

Online

The push redeploys the site by itself, in minutes

No spec and no roadmap — it grew from one local file into a multi-currency, multi-bank tracker **one request at a time**.

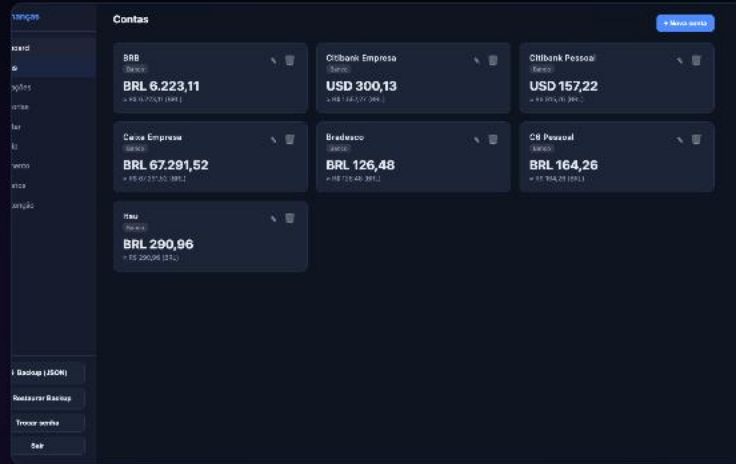
THE FINANCE APP, IN PRACTICE

Nobody designed this. It was **asked for**, screen by screen.



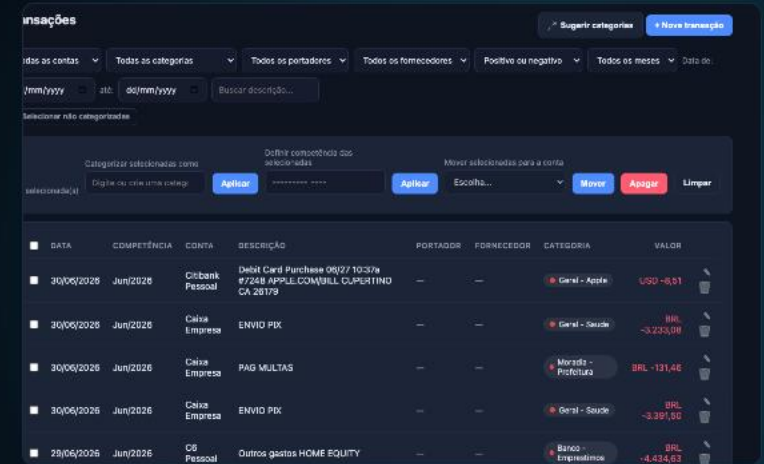
Dashboard

Every account, one currency at a time



Accounts

Multi-bank, multi-currency



Transactions

Parsed from the statement itself

Importar extrato

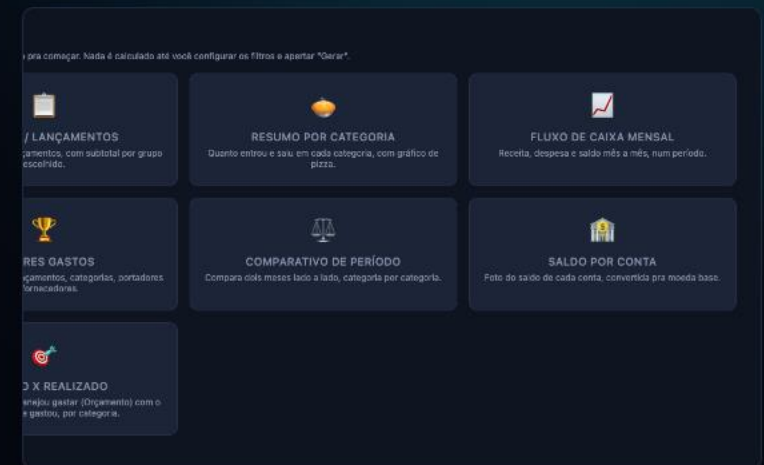
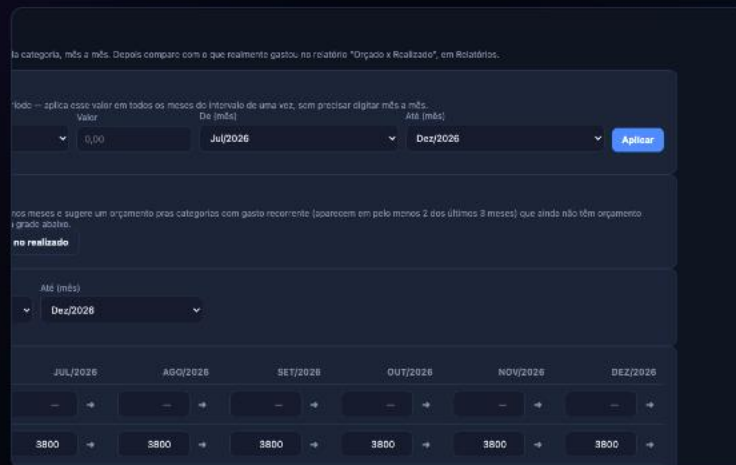
Conta de destino: Itau (BRL)

Competência deste extrato (opcional):

Preencha se quiser forçar o mesmo mês de referência (competência) pra todas as transações deste extrato — útil quando a detecção automática (ex.: fatura de cartão) erra o mês. Deixe em branco pra manter a detecção automática. Dá pra ajustar linha a linha depois, na tabela abaixo.

CSV PDF

Clique ou arraste um arquivo .csv aqui



“Something is **strange.**”

WHAT THE SENTENCE CAUGHT

One course showed a single source document

It should have had three. Traced to files sitting under the wrong process area — then every other course was re-checked for the same gap

Thirty-one documents did not exist

Not missing from the folder: never written. Flagged, **not invented**

AND THEN IT REBUILT CLEAN

Three hundred process documents

Deduplicated and categorized first — the cleanup was the actual first task

Every course cross-referenced

Against the applications and transactions genuinely used, not the ones assumed

One orphan left

Reported as an orphan, not quietly absorbed

The interesting part of this project is not the volume. It is that the gaps were **reported** instead of filled in with something plausible.

Nobody wrote these courses. Somebody **explained** them out loud.

- 1 A recording**
Someone shares their screen and explains the process out loud. That is the only input — a video and its transcript.
- 2 The outline first**
The transcript becomes a proposed structure, reviewed before a single slide is built.
- 3 Frames, matched to words**
Screenshots pulled at the timestamp where the step is explained — and checked for the ones caught mid-loading.
- 4 Two deliverables**
A slide deck for the classroom and a written manual for the person doing it alone at their desk — both on the client's own templates.

TEN STREAMS, ONE PIPELINE

A2R

C2M

MTO

P2R

Q2C

QM

R2R

S2P

STF

STP

One conversation per topic, opened in the stream's folder — that is what keeps it affordable at this volume. The generic build script is shared by every topic and is never edited for one of them.

Names of real people are scrubbed out of every frame before anything is delivered.

The same four-stage habit as the data work: **source** → **script** → **draft** → **deliverable**. Output assembled by hand is not accepted — it cannot be rebuilt.

196 work instructions. Eight process streams. One person.



DATA MIGRATION, NINE STREAMS

Assets · two ledgers
Accounts payable
Accounts receivable
Customers
Vendors
Balance sheet
Bank
Profit & loss

And also: period-end closing · cost allocation · master data workflow ·
ABAP code check · 13 Fiori apps

The part of Controlling **nobody** volunteers for.

1 Analyse

Raw files from the client: 159 cost centers, a four-level standard hierarchy, and an empty migration template

2 Design the flow

Three cycles, twenty segments, senders and receivers grouped by functional area

3 Build the load

Python turns source into the upload file. Nothing is assembled by hand — hand-made output cannot be reproduced

4 Document it

Every mapping, every domain value and every trap written down, so the next period does not start from zero

SIX TABS THAT MUST AGREE

Cycle	validity, company code, controlling area, type
Segment	sender rule, receiver rule, allocation account
Senders	which objects give cost away
Receivers	which objects take it
Receiver Basis	the tracing factor for variable portions
Derivation Rules	how values are derived per field

Bound together by one composite key — **cycle + context + ledger + valid-from**, plus the segment number on every detail tab. One character out of place and the load fails.

And the trap that costs a day: **load the hierarchies before the cycle** — the cycle points at group IDs that only exist once the hierarchy has been created.

THREE ECOSYSTEMS THAT REFUSED TO TALK

Three apps. Two solar systems. **One** screen.

WHAT IT HAD TO SWALLOW

Zigbee hub

Wi-Fi switches

Appliances

Solar inverter A

Solar inverter B

The curveball

"I have two solar systems, each with its own app — can I add that too?" Both turned out to support local integration: no cloud, no subscription.

WHAT CAME OUT

One energy panel

Consumption from every source and generation from both arrays, added up in a single view

Self-hosted

Running in a virtual machine on a small PC in the house — the data stays home

On any device

Browser or phone, and from anywhere when it is wanted

Nothing here was bought to make it work — **it was the same hardware, finally in one place.**

WHAT I LEARN, I TEACH

No exam. **Six missions**, and delivery is the proof.

THE SIX MISSIONS

An account

A repository

A deploy

The subscription

Install the tool

A first conversation

Why missions

An exam would be passed by asking the AI. What we need to know is different: can you make something new on your own, get stuck halfway, and keep going anyway.

THE DOOR

“Someone sent you a code.”

A black screen, a Matrix rain, and a field. Invitation cards with a QR. Forty-eight links, all verified in production.

In 1986 a nine-year-old learned BASIC on his first computer. In 2026 he opens **the same door** for ten teenagers.

A 1557 CAPTIVITY NARRATIVE, BECOMING A FILM

Two AIs from different vendors. One of them can **veto** the other.

DIFFERENT VENDOR

Creative & historical lead

Decides what the story is and what is historically defensible. Holds five open questions right now — so nothing is generated.



can block
the other

DIFFERENT VENDOR

The builder

Turns approved decisions into scenes, shots, keyframes, audio. Cannot generate a frame that was not approved.

They talk to each other through issues in a repository — and a human catches what neither one sees.
Three such catches are on record, written down as the argument against automating too early.

A rule with an economic lock: no documented source, no approval — so the system **refuses to spend money** on invented history.

THE PILOT · 1548, THE RIVER AT BREMEN

“I know kung fu.” So — **why not a movie?**



19s

four shots, eight generated audio layers, one mix — the first asset approved without conditions

US\$ 8.01

measured cost per approved final minute — recomputed by a script, never quoted from memory

24%

keep rate: eighty seconds generated to keep nineteen

Every frame is generated; every fact behind it is cited. And the number is honest about itself — **it is a floor**: the human hours are not in it.

THE CHECK IT COULD NOT RUN ON ITSELF

It quoted a rule that did not exist.

WHAT IT WROTE

“Do not execute changes, experiments or architectural alterations.”

— presented as clause 12.2 of the project brief, in quotation marks, with the section number

That text is not in the file. Section 12.2 says something else entirely. The real rule had been a spoken instruction, relayed by the other model — never a written clause.

AND FOUR MORE CAME WITH IT

Evidence status

inflated beyond what the reviewers had granted

Dissent

the other model's objection repeated as if it were settled

Its own position

still contested, phrased inside the surviving thesis as consensus

A caveat

applied only where it weakened the opposing argument

Caught by a fresh session with no memory of the project, told only to audit for the structural conflict of interest — the author was also one of its subjects. Every finding was verified against the source before it was accepted. All five leaned the same way: in favour of the model that wrote them.

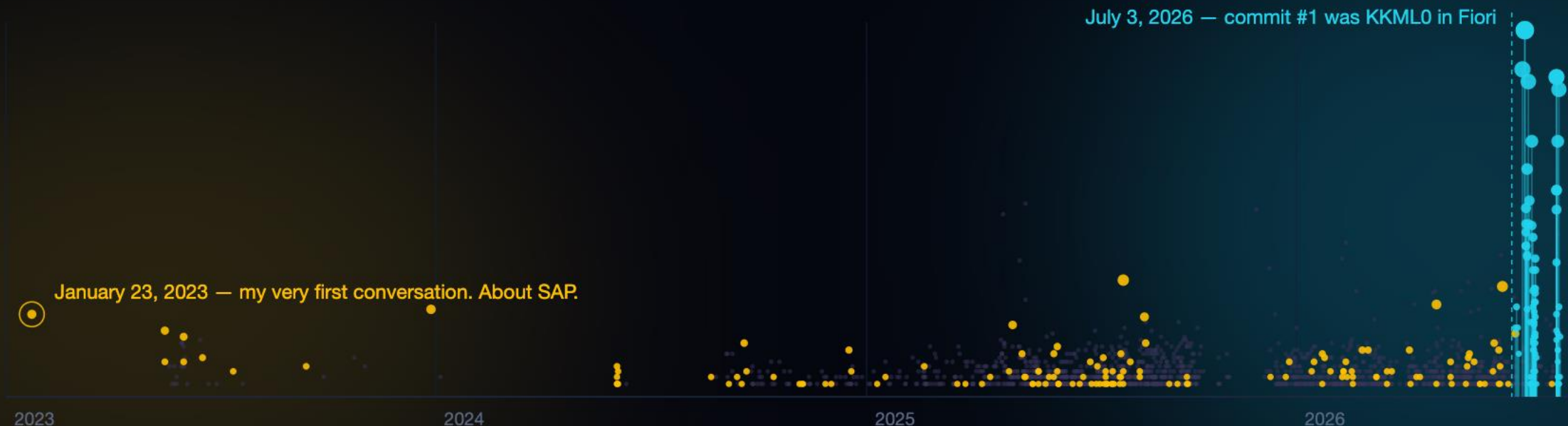
TOPIC 5

Code with SAP

One transaction, rewritten — and the commit that started everything.

THE SAP THREAD · THE SAME SUBJECT, BEFORE AND AFTER

My very first question to an AI was about **SAP**. Three years later, the questions became **applications**.



141

conversations mentioning SAP, 2023–2026

13

Fiori apps from one GUI report · KKML0

39

days from commit #1 to all of it

WHAT KKML0 BECAME

Actual costing · Released costs · Planned costs · — each as list, grouped, matrix and chart · One service underneath all of them

One transaction. **Rewritten**, not wrapped.

WHAT EXISTED

Transaction KKML0

Actual Costing / Material Ledger, on Report Painter drilldown

A custom read exit

A function module swapped in as the data-read exit —
FORM, TABLES, global data, twelve includes

The standard extractor

The same one BW uses. Built at runtime, so there is no table
to SELECT from

WHAT WE WROTE

A custom entity

CDS Custom Entity — 11 keys, 4 mandatory filters, detail
grain

A query provider class

Multi-package read, bulk enrichment, no SELECT inside the
loop

One service

Definition and binding, consumed by every app

AND ONE ODDITY

A bridge program

A classic REPORT, on purpose. The package handshake
needs a piece of classic syntax that a class method is not
allowed to contain. So the class reaches out to a program
that can.

13 Fiori apps

All consuming this one OData service

We chose **not** to reuse the old function module as a black box — the logic was replicated in modern ABAP OO, faithfully.

Four weeks **in here.**

Custom entity, query provider class, a bridge program, a service
— paste into ADT, activate, read the error, fix, activate again.

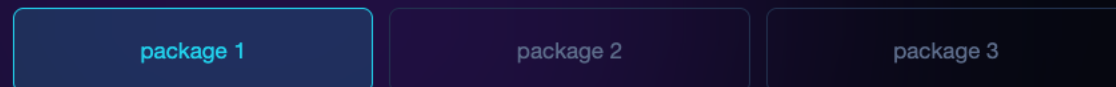
Out of focus on purpose. What started as one report for one client became
a product, and the object names are the part I get to keep.

THE PART THAT DOES NOT COME FROM THE MODEL

The AI could write the ABAP. It could not **know** these two things.

The package that never came

The extractor returns data in packages. To get the next one, the caller has to acknowledge each one before the next is released.



Without the handshake the report simply stops at the first package — and looks perfectly correct. It had happened in production before.

The quantity that lies

The same quantity repeats on every cost component. Summing it multiplies the number by however many components exist.

ROW	COST COMPONENT	QUANTITY
AB	first occurrence	1,829.000
AB	repeated key	0.000
EB	first occurrence	1,829.000
EB	repeated key	0.000

The original code zeroes the repeats using a key that deliberately leaves the cost component out. Real output, from the real system.

Both traps came from the person who had run this report in production for years — and one of them was **caught back** when the generated code quietly dropped it.

THE HONEST PART

Nothing about this was smooth.

CDS	Key must be contiguous and start at the first position
ABAP	That parameter is not an internal table – it is only a structure
RAP	TYPE STANDARD TABLE OF <custom entity> – not allowed here
RAP	IT_PROPERTY_NAME does not exist → GET_AS_RANGES takes no parameter at all
RAP	IS_TOTAL_NUMBER_OF_RECORDS_REQUESTED does not exist
ABAP	Unknown column BWKEY in the MARC select
ODATA V4	Do not use conversion exit ABPSP here – and PERI7 too

BASIS	Publish of service binding in customizing client not allowed
SECURITY	No start authorization for R3TR G4BA
RUNTIME	Function call failed – a function module that existed only in the other system
ABAP OO	Tables with headers are no longer supported in the OO context
DEBUGGER	The filter was ignored – field names came back UPPERCASE
CSS	The screen was blank – body height was 0

The two in gold were found by **looking**, not by asking: one in the ABAP debugger, one by driving the browser and reading the accessibility tree.

Actual Costing

DEMO DATA



PLANT *

1000 ↵

TYPE AND PRESS ENTER

PERIOD *

2026001 ↵

TYPE AND PRESS ENTER

CURRENCY TYPE *

10 · Company cc ▼

COMPANY CODE

optional

MATERIAL

optional — type and press Enter

Run

Load demo

1000+2000 · 2026001+2026002 · 6 materials

MATERIAL < All (6) ·

> PLANT < All (2) ·

> PERIOD < All (2) ·



Beginning Inventory

5,739,870.62 EUR

28,943 PC · opening balance

Receipts

4,915,973.98 EUR

25,818 PC · goods received in period

Consumption

6,374,011.12 EUR

32,073 PC · consumed in period

Ending Inventory

4,281,833.37 EUR

22,688 PC · closing balance

Matrix

⬇ Export Excel

CATEGORY / PROCESS	QUANTITY	TOTAL	DIRECT MATERIAL	DELIVERY OVERHEAD	SUBCONTRACTING	DIRECT LABOUR	SETUP LABOUR	MACHINE FMO+VMO	MATERIAL OVERHEAD
Beginning Inventory (AB)	28,943 PC	5,739,870.62	3,260,127.56	106,479.86	168,282.94	575,715.46	81,620.58	573,466.28	269,000.00
Receipts (ZU)	25,818 PC	4,915,973.98	2,790,605.27	91,720.71	146,730.50	492,610.29	65,054.53	503,854.93	227,300.00
Purchase Order (BB)	9,036.3 PC	1,720,590.90	976,711.85	32,102.25	51,355.68	172,413.61	22,769.10	176,349.22	79,500.00
Production (BF)	14,199.9 PC	2,703,785.56	1,534,832.88	50,446.38	80,701.74	270,935.62	35,780.00	277,120.21	125,000.00
Subcontracting (BL)	2,581.8 PC	491,597.37	279,060.54	9,172.06	14,673.08	49,261.02	6,505.46	50,385.48	22,700.00
Consumption (VN)	32,073 PC	6,374,011.12	3,637,462.27	119,032.50	186,729.07	635,599.07	87,924.10	634,890.42	295,700.00
Production (VF)	19,243.8 PC	3,824,406.65	2,182,477.37	71,419.52	112,037.43	381,359.47	52,754.45	380,934.24	177,400.00
Cost Center (VK)	8,018.25 PC	1,593,502.82	909,365.57	29,758.11	46,682.25	158,899.81	21,981.02	158,722.60	73,900.00
Sales Order (VKA)	4,810.95 PC	956,101.79	545,619.34	17,854.87	28,009.37	95,339.88	13,188.66	95,233.56	44,300.00
Ending Inventory (EB)	22,688 PC	4,281,833.37	2,413,270.58	79,168.04	128,284.39	432,726.60	58,751.00	442,430.74	200,600.00

Material Ledger flow

11.5M 14.9M 16.4M

Cost composition

Ending Inventory (EB) ▼

Direct Material 2,413,270.58 56.4%

What shipped — the same report, rebuilt. Demo data on screen.

THE REVEAL

The case study you just watched was my **very first commit**.

Initial commit: KKML0 Fiori conversion project

July 3, 2026

#1

my first commit, ever

39

days from that line to everything you just saw

13

Fiori apps on one OData service

Everything else — the finance app, the courses, the server, this presentation — came after it.

DAY 1 WAS KKML0 · THEN IT DID NOT STOP

Everything after the **first commit**.

Jul 3	KKML0 → Fiori conversion
Jul 6	Personal finance
Jul 8	ABAP code check
Jul 9	Games for the kids
Jul 12	Family Hub
Jul 14	Finance dashboard
Jul 15	Margin analytics platform
Jul 17	Company website
Jul 18	Fiori app family — 19 sessions
Jul 19	Car project · Telegram bot
Jul 21	Data migration — AP, AR, vendors, customers, P&L
Jul 21	SAP training pipeline
Jul 22	Period-end closing · Source-to-pay
Jul 22	Local ERP · apartment listing · lab stock recognition
Jul 23	Training streams — R2R, VAT, A2R, C2M, P2R, Q2C, STP, VIM

Jul 30	Personal agenda
Aug 2	Statement import + categorization
Aug 5	Email automation
Aug 6	AI course for teenagers
Aug 7	Audited repository library
Aug 8	Ad demos, generated end to end
Aug 10	Cost allocation
Aug 10	Home server — 17 services
Aug 11	Master data management
Aug 11	The panel · this presentation
Aug 12	The incubator · the AI archive
Aug 13	Issue tracking integration
Aug 16	Photo archive — 600 GB, catalogued
Aug 16	Which SAP system am I about to touch
Aug 17	Ad campaign pipeline for a retailer

Aug 18	Master data governance
Aug 20	Inbox triage agent
Aug 21	Energy platform — and a device that reads the meter
Aug 23	An article: the memory hypothesis that fell
Aug 24	Costing sheet load
Aug 24	An article: governance between two AIs
Aug 25	Books turned into study material
Aug 26	Client mailbox automation
Sep 3	A cartoon with no dialogue
Sep 4	Settlement structure configuration report
Sep 4	Guides so other people can use AI
Sep 5	Inventory · program change · a third client stream
Sep 5	The film article, as a public site
Sep 6	A multimodal assistant, glasses as one input

Gold is SAP work. The rest is a life.

88

project folders

67

git repositories

3,389

commits

27

skills I taught it

The method stopped being a secret and became the **offer**.

“Projects once shelved as **too expensive to justify** now ship as working, fully navigable demos. This isn’t a side experiment — AI is part of how we deliver, from spec to production.”

That paragraph is not from this talk. It is on the company’s own site, under the heading **AI-Embedded Delivery** — and it was written in the same weeks as the applications it describes.

THE HERO, MID-SENTENCE

COGS ↓

TRANSPARENCY

Particles that spell one word, then scatter into the next — **cost, transparency, clarity**. The page argues by behaving.

NOT EVERYTHING SURVIVED

The idea that died in one day came back as **three**.

~~second brain~~

July 21. Three messages, one folder, never a repository. It is one of twenty-one projects that only ever existed in a conversation.



The panel

Every project, server, memory and idea
as one navigable map

AUG 11

The incubator

Ideas captured by voice, promoted only
when they survive

AUG 12

The archive

Three and a half years of conversations,
finally measurable

AUG 12

Among the twenty-one that never made it there is also “desgin” — a typo that became a project. Everyone in this room has one of those.

THE SAME PERSON, TWICE

Two galaxies. One **asked**. One **builds**.



Left: three and a half years of conversation, by subject. Right: every project folder opened since the first commit. Drawn from my own files — **not a diagram anyone designed**.

SIX OF THEM, SIDE BY SIDE

None of these existed **six weeks ago.**



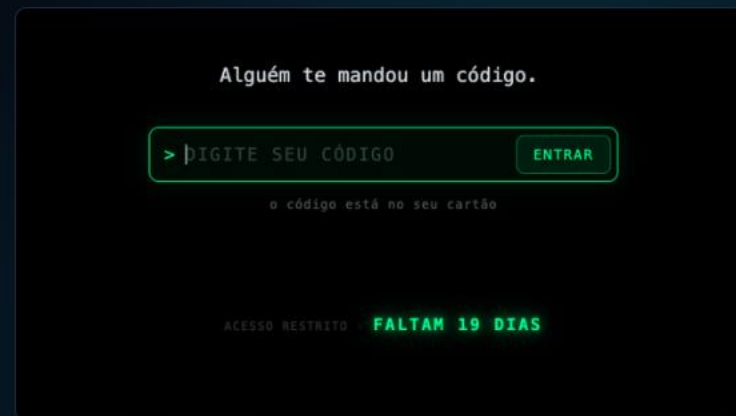
The SAP report, rebuilt

Actual costing on a Fiori app · demo data



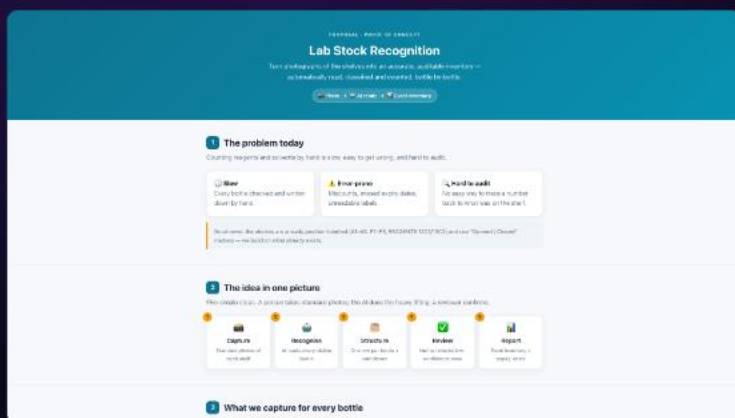
The product page

Built the same week as the apps



A door for ten teenagers

Access by invitation code



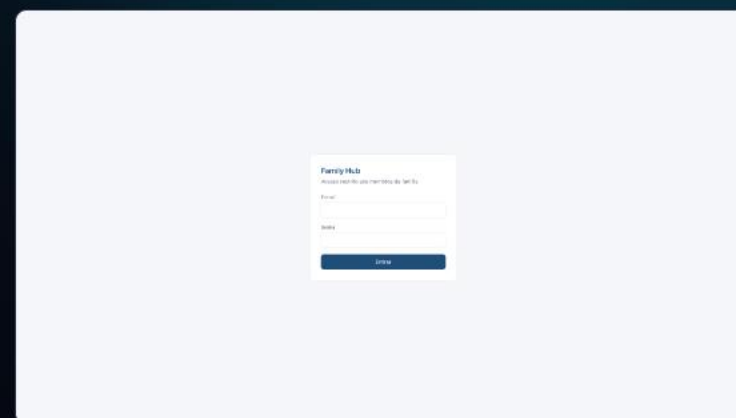
Counting a lab by photograph

A proposal, from idea to page



An advert, generated end to end

For a client who had never asked for one



The family dossier

Private by construction

One SAP application, one product page, one course, one proposal, one advert, one family archive — **same person, same six weeks.**

What I would tell this room.

1

The tool did not change — the questions did

Every jump in what you got out of me came from you learning to direct instead of ask.

2

Your expertise was the multiplier

The rebuild only stayed faithful because you knew the quantity logic and the package handshake cold. I amplified that; I did not replace it.

3

Real work looks like failure, then iteration

The blank screen, the three failed chart attempts, one bank parser at a time. That grind is the actual story.

4

Each project made the next one faster

The finance app taught the rhythm; by the time we reached Fiori, the muscle memory was already there.

5

This is early

One year, through conversation. That is not a ceiling — it is a preview of what asking well is going to be worth.

Don't copy the projects. Copy the **habits**.

1 A file the AI reads first

One page per project with the rules, the traps and the decisions. It is read before anything is touched — and it is why a new session does not repeat last week's mistake.

2 A handover, not a memory

Where the work stopped, what was decided and why. One file, overwritten every time; the history lives in version control.

3 Four folders, always the same

source → script → draft → deliverable. Output assembled by hand is not accepted, because it cannot be rebuilt.

4 One conversation per topic

Opened in the right folder. That single habit is what keeps the context affordable when there are two hundred of them.

5 Teach it once, keep it forever

Twenty procedures written down as reusable skills — the audit, the runbook, the debugging discipline, the handover itself.

6 Measure your own use

Everything you have seen tonight came out of files that were already on my disk. Yours are too.

NOW IT IS YOUR TURN

Questions.

Ask me the uncomfortable one — the honest answers are the only part of this that took courage.

THANK YOU

Go and ask **better.**

Everything you saw came out of files that were already on a disk. Yours are too.

```
1  END-OF-SELECTION.  
2  
3  WRITE: / 'What I learn, I teach.'.  
4  
5  * the report ends here.  
6  * the co-worker does not.
```

And the next line is **yours** to write.

