“Quien sabe por Algebra, sabe scientificamente”

Lição — J.N. Oliveira

Universidade do Minho, 14/11/2025

Saudação

Do soldado **88**
ao soldado **187**

Agradecimentos

A todos os colegas e funcionários com quem lidei.

Dedicatória

A todos os meus alunos.

"Quien sabe por Algebra, sabe scientificamente"

(Pedro Nunes, *Libro de Algebra en Arithmetica y Geometria*, 1567, fol. 270)

Ou antes...?

"Não sei por onde vou,
Não sei para onde vou
— Sei que não vou por aí!"

(José Régio, Cântico Negro)

Problema

Os meus três filhos nasceram com três anos de diferença. Juntos, têm a mesma idade que eu, 48 anos. Quantos anos têm eles?

Problema

Os meus três filhos nasceram com três anos de diferença. Juntos, têm a mesma idade que eu, 48 anos. Quantos anos têm eles?

Modelo formal

$$x + (x + 3) + (x + 6) = 48$$

— descrição **matemática** do problema.

‘Calculemus‘

$$
\begin{aligned}
& 3x + 9 = 48 \\
\equiv & \quad \{ \text{ regra "al-djabr" } \} \\
& 3x = 48 - 9 \\
\equiv & \quad \{ \text{ regra "al-hatt" } \} \\
& x = 16 - 3 \\
\equiv & \quad \{ \text{ aritmética } \} \\
& x = 13
\end{aligned}
$$

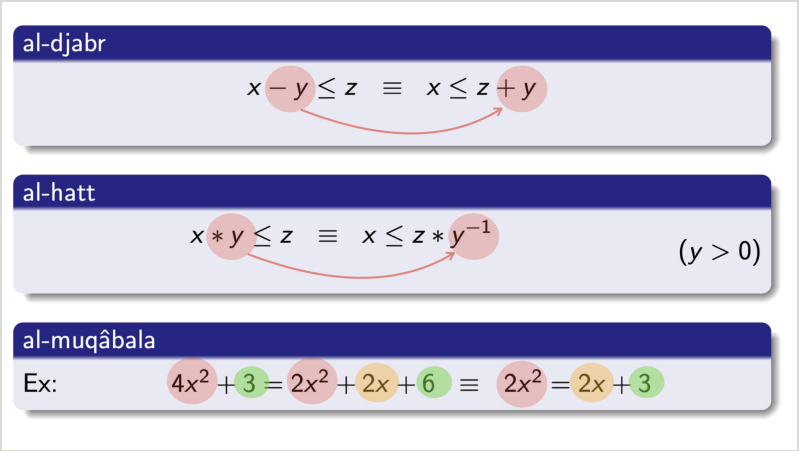
‘Calculemus‘

$$
\begin{aligned}
& 3x + 9 = 48 \\
\equiv & \quad \{ \text{ regra "al-djabr" } \} \\
& 3x = 48 - 9 \\
\equiv & \quad \{ \text{ regra "al-hatt" } \} \\
& x = 16 - 3 \\
\equiv & \quad \{ \text{ aritmética } \} \\
& x = 13
\end{aligned}
$$

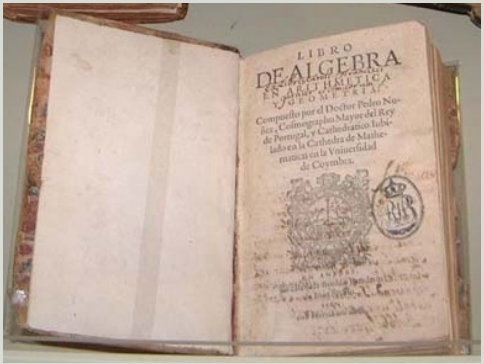
Solução

$$
\begin{aligned}
x &= 13 \\
x + 3 &= 16 \\
x + 6 &= 19
\end{aligned}
$$

”Al-djabr”? ”al-hatt”? (”al-muqâbala”)?



Libro de Algebra en Arithmetica y Geometria (1567)



(...) ho inuẽtor desta arte foy hum Mathematico Mouro, cujo nome era Gebre, & ha em alguãs Liuarias hum pequeno tractado Arauigo, que contem os capitulos de q̃ usamos (fol. a ij r)

Referência a **On the calculus of al-gabr and al-muqâbala** [1] por Abû Abd Allâh Muhamad B. Mûsâ **Al-Huwârizmî**, famoso matemático persa do séc. IX.

[1] Título original: *Kitâb al-muhtasar fi hisab al-gabr wa-almuqâbala*.

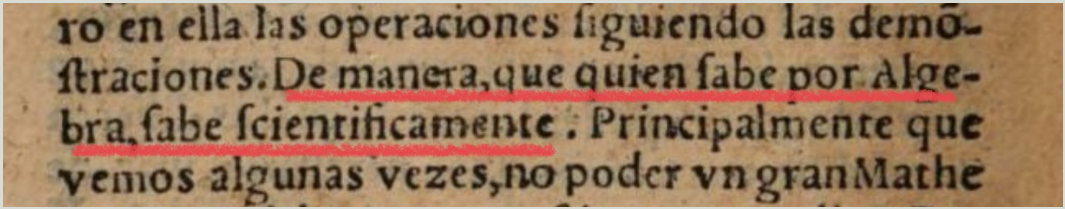
Libro de Algebra en Arithmetica y Geometria (1567)

Fols. 270–270v:

> *(...) Principalmente que vemos algumas vezes, no poder vn gran Mathematico resoluer vna question por medios Geometricos, y resolverla por Algebra, siendo la misma Algebra sacada de la Geometria, q̃ es cosa de admiracio̅.*

Libro de Algebra en Arithmetica y Geometria (1567)

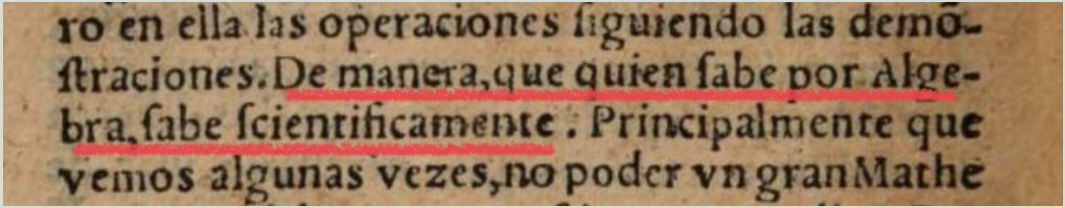
Fol. 270v:



ro en ella las operaciones ſiguiendo las demõ-
ſtraciones. De manera, que quien ſabe por Alge-
bra, ſabe ſcientificamente: Principalmente que
vemos algunas vezes, no poder vn gran Mathe

Libro de Algebra en Arithmetica y Geometria (1567)

Fol. 270v:


ro en ella las operaciones ſiguiendo las demõ-
ſtraciones. De manera, que quien ſabe por Alge-
bra, ſabe ſcientificamente: Principalmente que
vemos algunas vezes, no poder vn gran Mathe

Motivação: Quão actual é, nos dias de hoje, este juízo de Nunes?

1963/73 — “Matemáticas Modernas”

José Sebastião e Silva

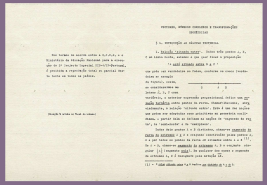
Contexto histórico

Desde o final da 2ª Guerra Mundial que se assiste a uma valorização do conhecimento científico como factor simultaneamente de progresso humano e de hegemonia estratégica na maioria dos países europeus, no continente americano e nos países do Bloco de Leste.

A matemática é vista com particular atenção e na primeira metade dos anos 50 começa a tomar forma o movimento que constituirá o que se costuma designar por Matemática Moderna.

Portugal, através de José Sebastião e Silva (1914–1972), esteve desde muito cedo (1952) em contacto com este movimento.

A experiência da Matemática Moderna nos Liceus



Compêndio de matemática (3º volume – 7º ano)
José Sebastião e Silva. 1ª Edição 1965

Páginas 2–3

Compêndio de matemática
1º ano do ensino liceal (antigo 3º ano)
António Almeida e Costa e Alfredo Dinis dos Anjos
1ª Edição 1972

Capa

Compêndio de matemática
Curso Complementar 1º ano, 1º volume
Maria Helena Garcia, Alfredo Dinis dos Anjos, António Fernando Reino 1ª Edição 1973

Capa

A Matemática Moderna nos liceus

Com o apoio do ministro Leite Pinto vai-se iniciar no ano lectivo de 1963-64 numa experiência de ensino envolvendo uma turma do 4º ano em cada um dos liceus normais (Coimbra, Lisboa, Porto) constituída com os melhores alunos de cada liceu.

A pouco e pouco vão crescendo o número de turmas e de professores envolvidos. A experiência é generalizada em 1973/74. Sebastião e Silva publica cinco livros para esta experiência (três livros de texto e dois guias para o professor) entre 1964 e 1969.

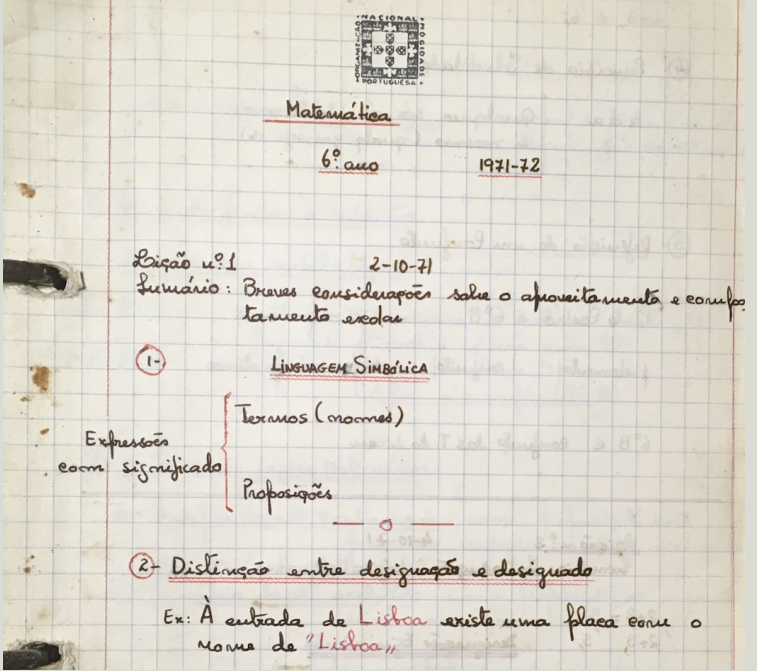
livros de texto de matemática

20 ANOS

Associação de Professores de Matemática

1971/72

Liceu Nacional da Póvoa de Varzim, há 54 anos...



Matemática

6º. ano 1971-72

Lição nº 1 2-10-71

Sumário: Breves considerações sobre o aproveitamento e comportamento escolar

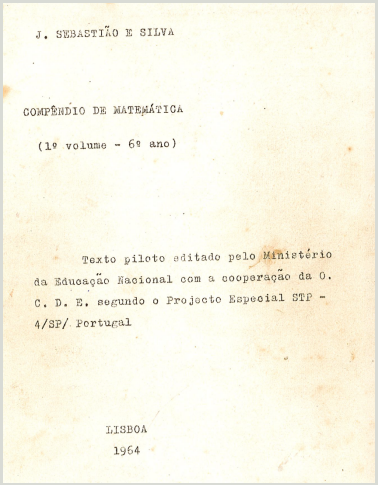
1- Linguagem Simbólica

Expressões com significado
- Termos (nomes)
- Proposições

2- Distinção entre designação e designado

Ex: À entrada de Lisboa existe uma placa com o nome de "Lisboa"

1971/72 — J. Sebastião e Silva



J. SEBASTIÃO E SILVA

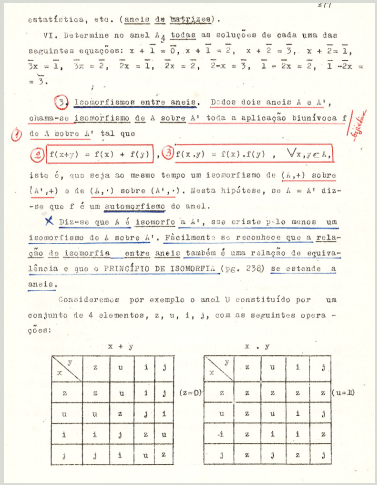
COMPÊNDIO DE MATEMÁTICA

(1º volume - 6º ano)

Texto piloto editado pelo Ministério da Educação Nacional com a cooperação da O. C. D. E. segundo o Projecto Especial STP - 4/SP/ Portugal

LISBOA

1964



estatística, etc. (anéis de matrizes).

VI. Determine no anel A_4 todas as soluções de cada uma das seguintes equações: $x + \overline{1} = \overline{0}$, $x + \overline{1} = \overline{2}$, $x + \overline{2} = \overline{3}$, $x + \overline{2} = \overline{1}$, $\overline{2}x = \overline{1}$, $\overline{3}x = \overline{2}$, $\overline{2}x = \overline{1}$, $\overline{2}x = \overline{2}$, $\overline{3}-x = \overline{3}$, $\overline{1} - \overline{2}x = \overline{2}$, $\overline{1} - \overline{2}x = \overline{3}$.

3. Isomorfismos entre anéis. Dados dois anéis A e A', chama-se isomorfismo de A sobre A' toda a aplicação bijectiva f de A sobre A' tal que

(2) $f(x+y) = f(x) + f(y)$, (3) $f(x.y) = f(x).f(y)$, $\forall x,y \in A$,

isto é, que seja ao mesmo tempo um isomorfismo de $(A,+)$ sobre $(A',+)$ e de $(A,\cdot)$ sobre $(A',\cdot)$. Nesta hipótese, se A = A' diz-se que f é um automorfismo do anel.

Diz-se que A é isomorfo a A', se existe pelo menos um isomorfismo de A sobre A'. Fàcilmente se reconhece que a relação de isomorfia entre anéis também é uma relação de equivalência e que o PRINCÍPIO DE ISOMORFIA (pg. 256) se estende a anéis.

Consideremos por exemplo o anel U constituído por um conjunto de 4 elementos, z, u, i, j, com as seguintes operações:

x + y

x \ y	z	u	i	j
z	z	u	i	j
u	u	z	j	i
i	i	j	z	u
j	j	i	u	z

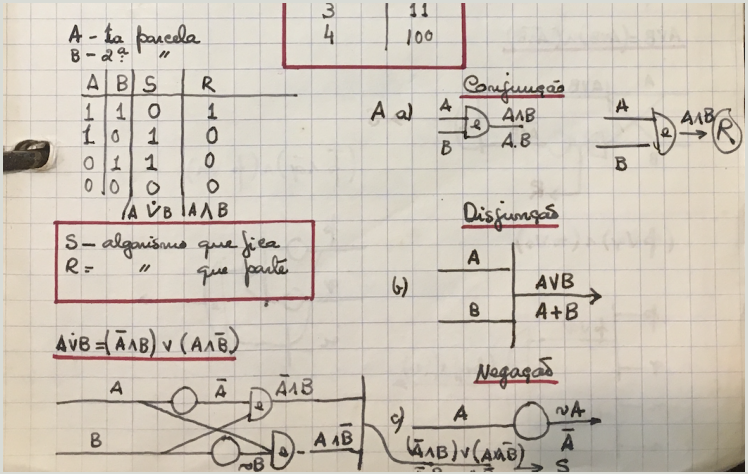
(z=0)

x . y

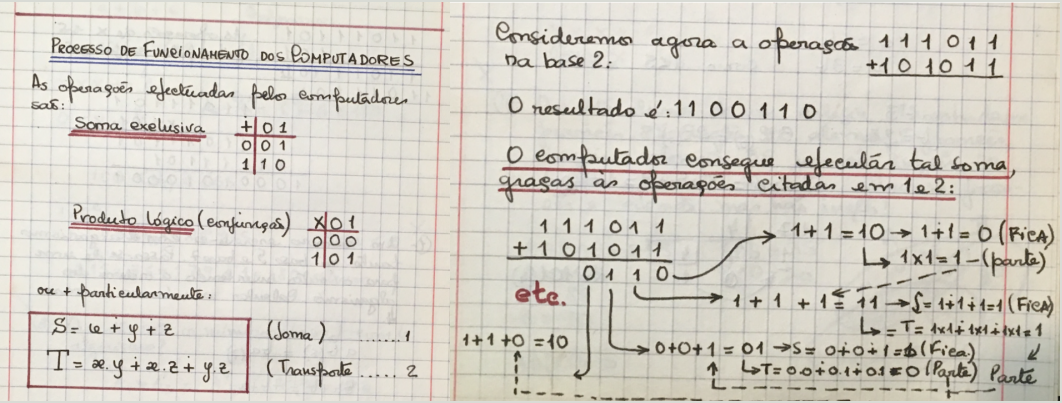
x \ y	z	u	i	j
z	z	z	z	z
u	z	u	i	j
i	z	i	i	z
j	z	j	z	j

(u=1)

1971/72 — circuitos lógicos



1971/72 — "(...) funcionamento dos computadores"



PROCESSO DE FUNCIONAMENTO DOS COMPUTADORES

As operações efectuadas pelos computadores são:

Soma exclusiva

+	0	1
0	0	1
1	1	0

Produto lógico (conjunção)

x	0	1
0	0	0
1	0	1

ou + particularmente:

$S = x + y + z$ (Soma) 1

$T = x.y + x.z + y.z$ (Transporte 2

Consideremos agora a operação na base 2:

1 1 1 0 1 1
+ 1 0 1 0 1 1

O resultado é: 1 1 0 0 1 1 0

O computador consegue efectuar tal soma, graças às operações citadas em 1 e 2:

1 1 1 0 1 1
+ 1 0 1 0 1 1
0 1 1 0

1+1 = 10 → 1+1 = 0 (Fica)
↳ 1x1 = 1 (parte)

etc.

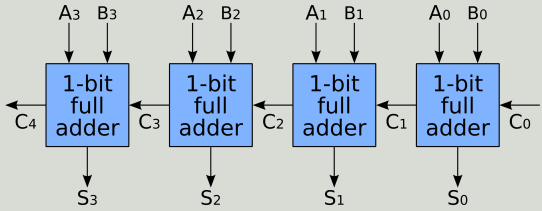
1 + 1 + 1 = 11 → S = 1+1+1 = 1 (Fica)
↳ T = 1x1 + 1x1 + 1x1 = 1

1+1+0 = 10

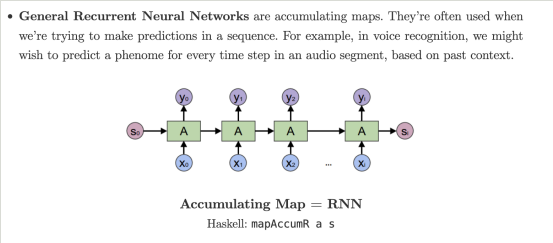
0+0+1 = 01 → S = 0+0+1 = 1 (Fica)
↳ T = 0.0 + 0.1 + 0.1 = 0 (Parte)

Parte

G.Stibitz, 1937



Talvez a “primeira rede neuronal” (**RNN**) de sempre?



(C. Olah's *Neural Networks, Types, and Functional Programming*, 2025.)

Anos 30 — década “vintage”

1936

A. Turing — noção abstrata do que chamamos hoje um **computador programável** *— conhecida por* **máquina de Turing**.

1936

A. Church — cálculo-λ, a base teórica da programação **funcional**.

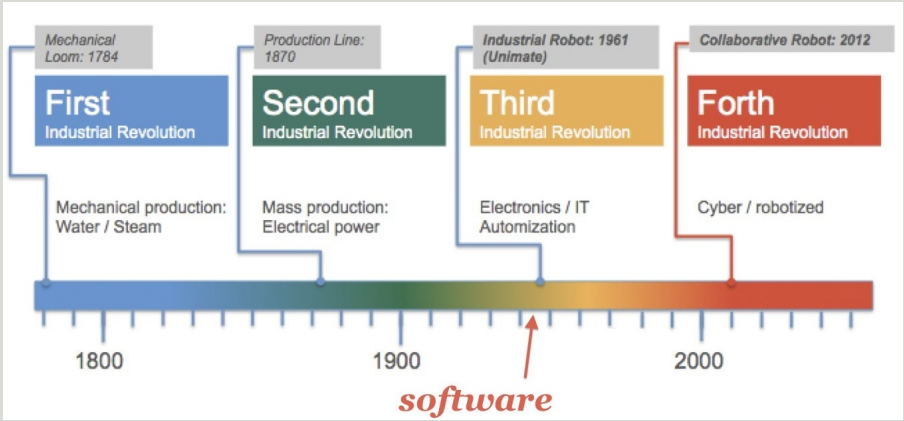
1937

*Primeiro computador digital — Atanasoff–Berry computer (***ABC***)*

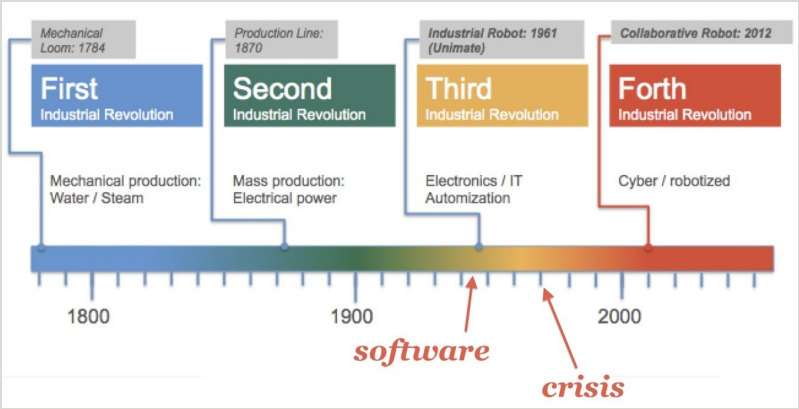
A. Turing (1912-1954)

Anos 40

Nasce *"l'enfant terrible"*



Anos 60 — “crise” no software



1st NATO Conf. on Software Engineering, Darmstadt, Oct. 1968

“L’enfant terrible”

Hardware — como outros **produtos** industriais "tradicionais", é fabricado segundo as leis da física.

Software — não parece ser regido pelas leis da física...

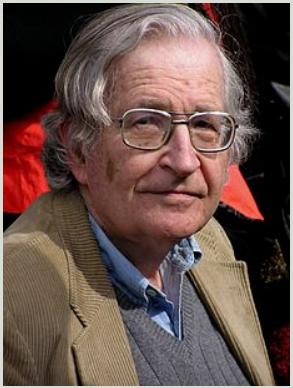
Anthony Oettinger (ACM President, 1967):

> *"(...) the scientific, rigorous component of computing, is more like* **mathematics** *than it is like* **physics"**.

Linguística

Reconhecendo não ser um produto da engenharia "tradicional", o **software** refugia-se na **linguística**:

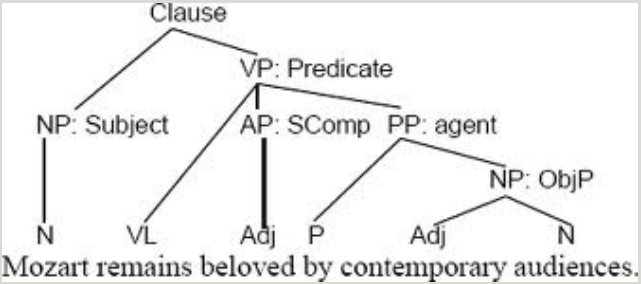
- **Gramáticas** generativas
- Tradução dirigida pela **sintaxe**
- significado do **todo** = *função* dos significados das **partes**



Noam Chomsky (1928-)

Contribuição da linguística

Essência de uma **gramática generativa**:

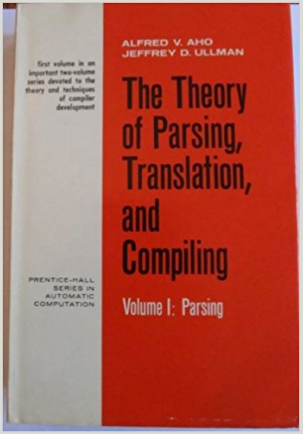


Computador "entende" frases sempre que consegue construir **derivações** como a da figura.

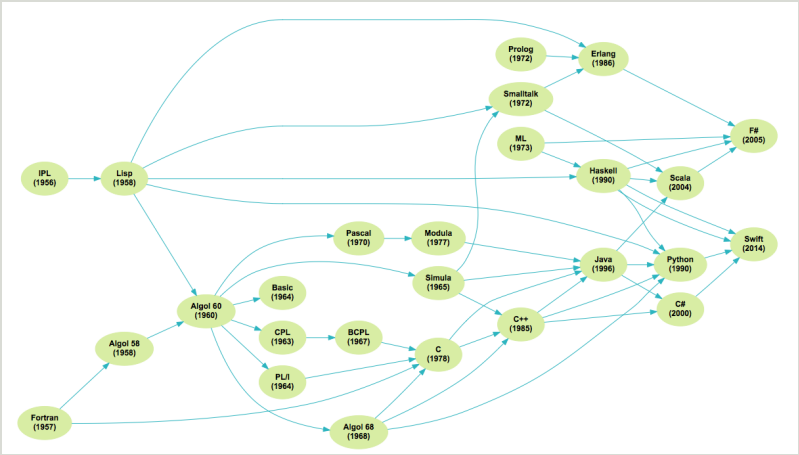
Anos 70

Teoria das **linguagens** consolida-se.

Aparecem as primeiras "**bíblias**", e.g.



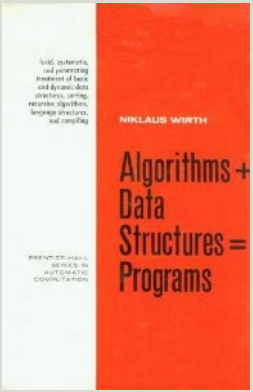
Explosão de linguagens

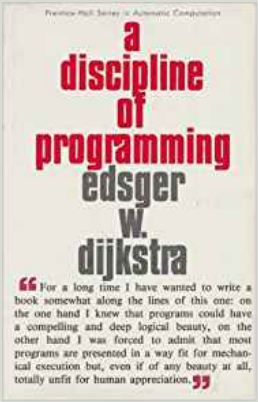


Anos 70 — década “vintage”

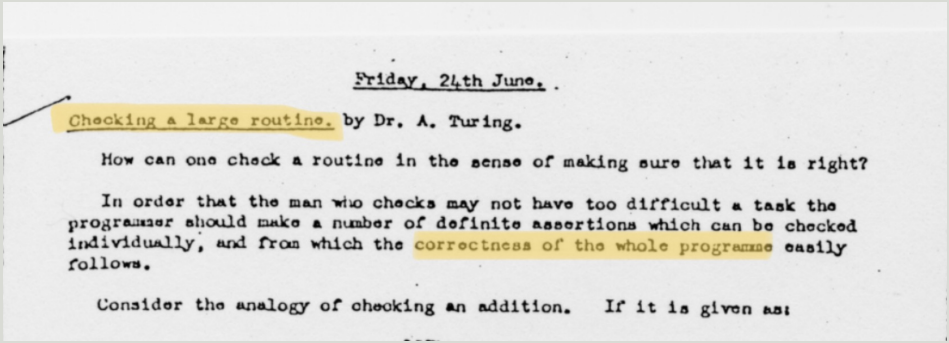
Surgem as primeiras bases teóricas da **programação** (dita ’estruturada’).

Mais livros de leitura obrigatória (1976).





Antecedentes (1949, A. Turing)



Friday, 24th June.

Checking a large routine. by Dr. A. Turing.

How can one check a routine in the sense of making sure that it is right?

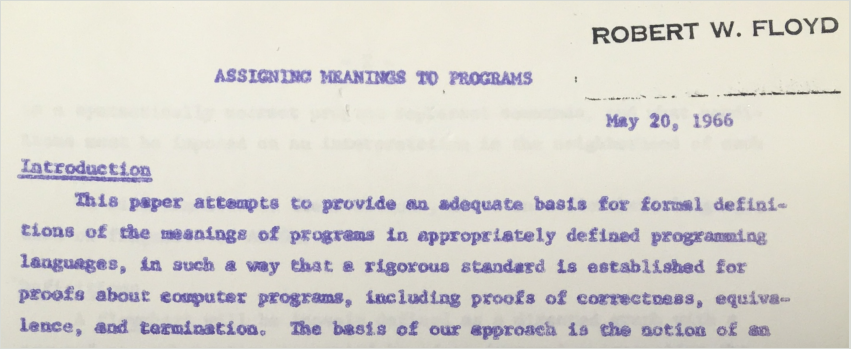
In order that the man who checks may not have too difficult a task the programmer should make a number of definite assertions which can be checked individually, and from which the correctness of the whole programme easily follows.

Consider the analogy of checking an addition. If it is given as:

Cliff Jones, 1984: *"(...) Sadly his paper had little impact. Understanding the problem faced, Turing's proposal (...) clarifies a* problem that still costs society a fortune each year."

Antecedentes (1966)

Floyd: o que "significa" um programa?



ROBERT W. FLOYD

ASSIGNING MEANINGS TO PROGRAMS

May 20, 1966

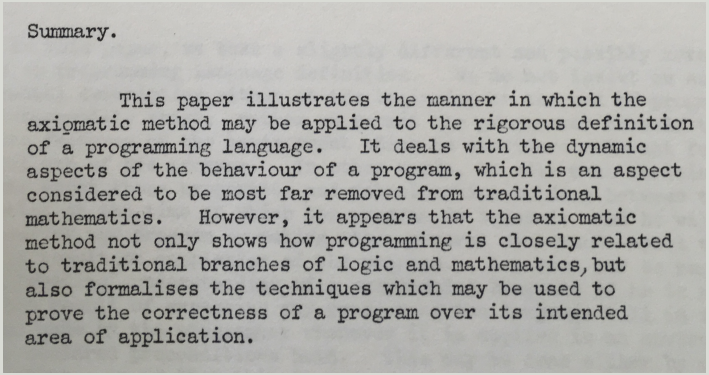
Introduction

This paper attempts to provide an adequate basis for formal definitions of the meanings of programs in appropriately defined programming languages, in such a way that a rigorous standard is established for proofs about computer programs, including proofs of correctness, equivalence, and termination. The basis of our approach is the notion of an

(TC2-WG2.1/004/01456)

Antecedentes (1968)

Lógica de Floyd-Hoare para provar a **correcção de programas**:



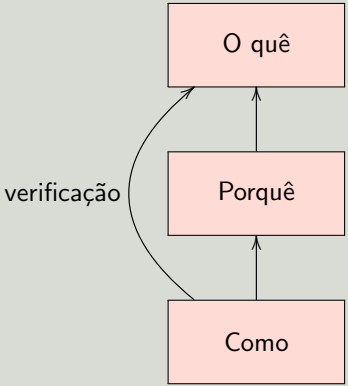
Summary.

This paper illustrates the manner in which the axiomatic method may be applied to the rigorous definition of a programming language. It deals with the dynamic aspects of the behaviour of a program, which is an aspect considered to be most far removed from traditional mathematics. However, it appears that the axiomatic method not only shows how programming is closely related to traditional branches of logic and mathematics, but also formalises the techniques which may be used to prove the correctness of a program over its intended area of application.

(ADB/IFIP/1164;1456)

“Métodos formais” — anos 70+

Formalização do **processo**:



especificação (*encomenda*)

justificação (**qualidade**)

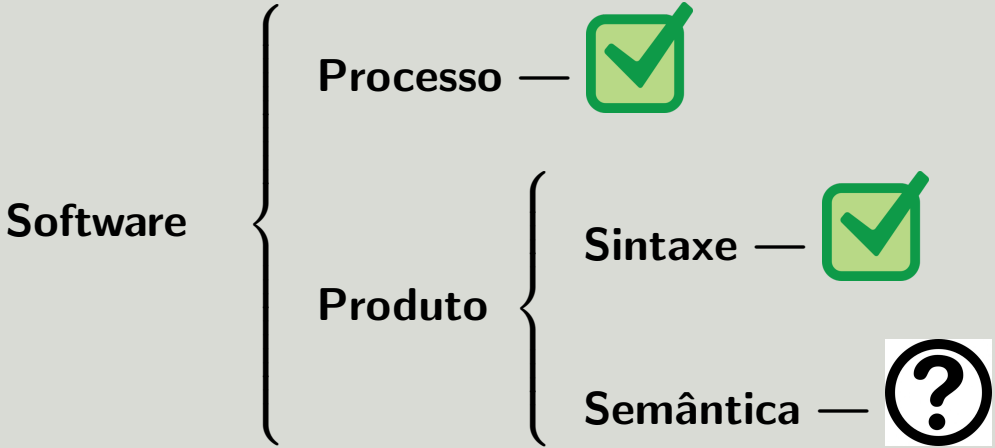
implementação (*programa* = **produto**)

Mas...

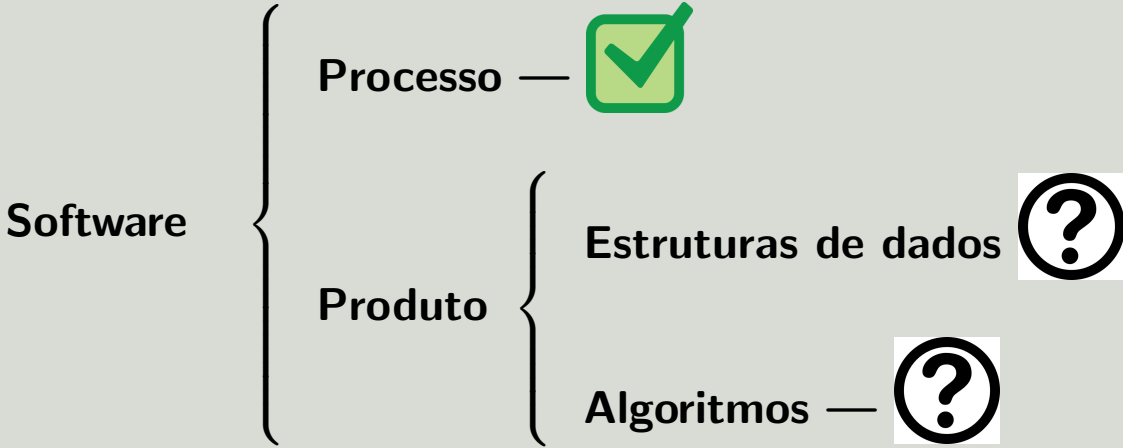
Software

- **Processo** — 
- **Produto** — 

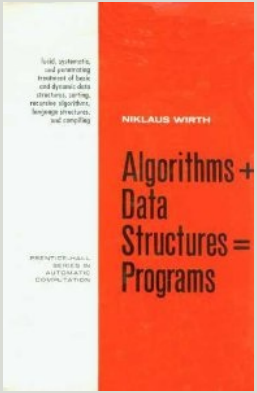
Abordagem linguística



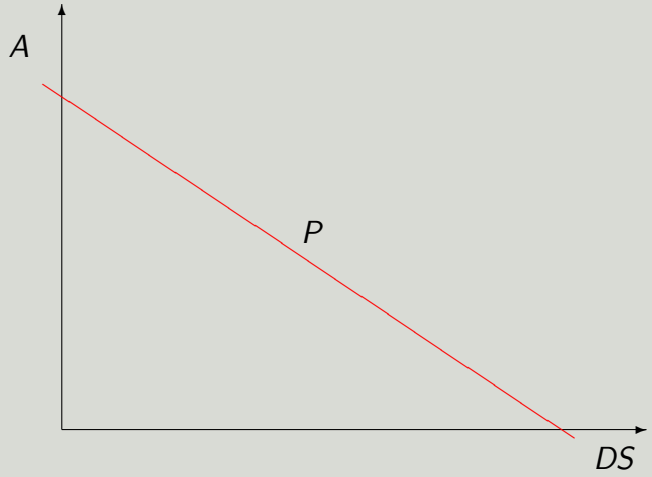
Abordagem estrutural



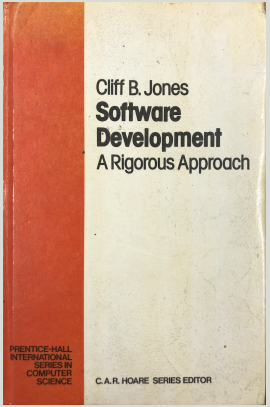
1976 — Niklaus Wirth (1934-2024)



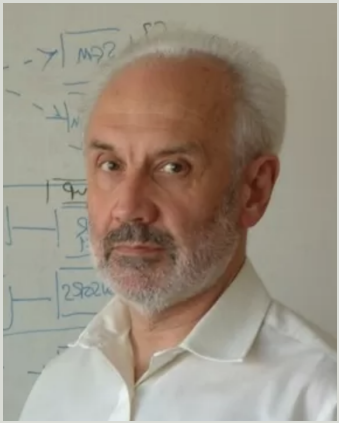
The $A + DS = P$ "equation"



1980 — Linguagens de especificação



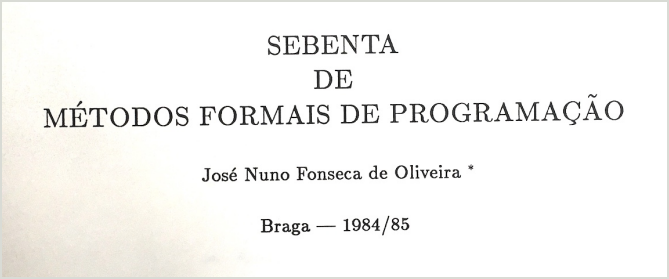
VDM — Vienna Development Method



Cliff Jones (1944-)

UM, 1984/85

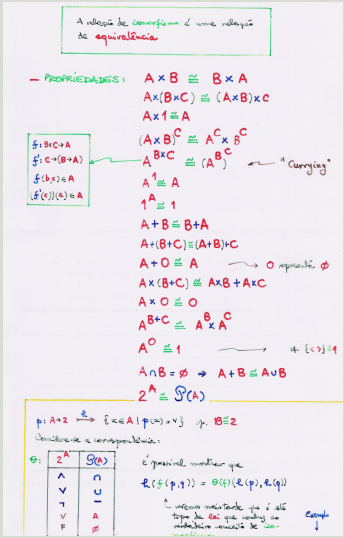
Opção 4º ano LESI (Anual):



SEBENTA
DE
MÉTODOS FORMAIS DE PROGRAMAÇÃO

José Nuno Fonseca de Oliveira *

Braga — 1984/85



UM, 1984/85

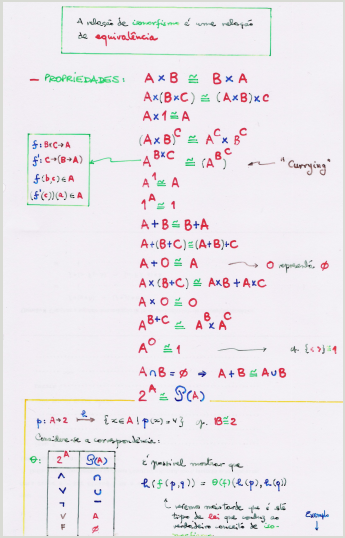
Opção 4º ano LESI (Anual):

SEBENTA
DE
MÉTODOS FORMAIS DE PROGRAMAÇÃO

José Nuno Fonseca de Oliveira *

Braga — 1984/85

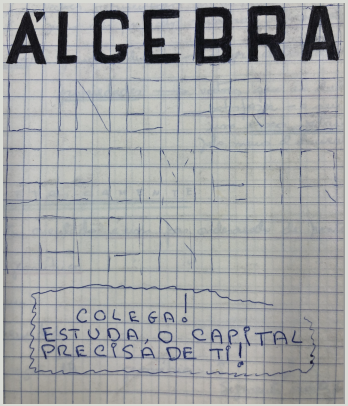
Antecedentes?



FEUP — 1973-1978

Eng.ª Electrotécnica — Sistemas Digitais e Computadores

FEUP — 1973-1978



Eng.ª Electrotécnica — Sistemas Digitais e Computadores

1978, Julho — Braga

1978, Julho — Braga

Laboratório de Electrónica

Prof. Raul Vidal

(Pavilhões da Rodovia)

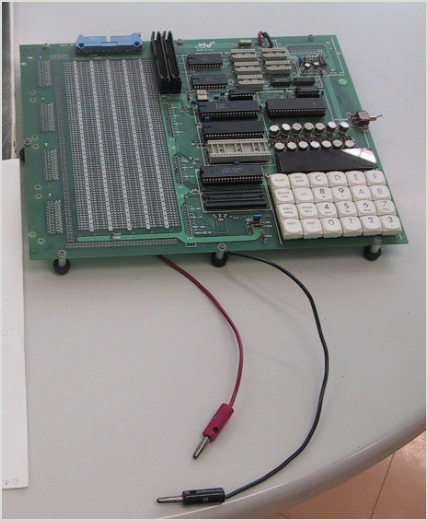
1978–80: UM, Laboratório de Electrónica

Tempos inesquecíveis.

Até convencemos um **SDK85** da Intel (+2Kb RAM) a tocar JS Bach a 4 partes.

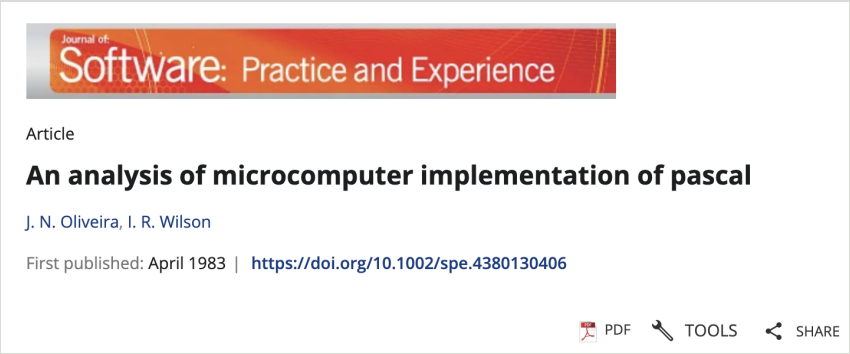
Tudo em código máquina (fita perfurada).

Contribuição para a Exposição JUVEMINHO'79 — Braga, Outubro de 1979.

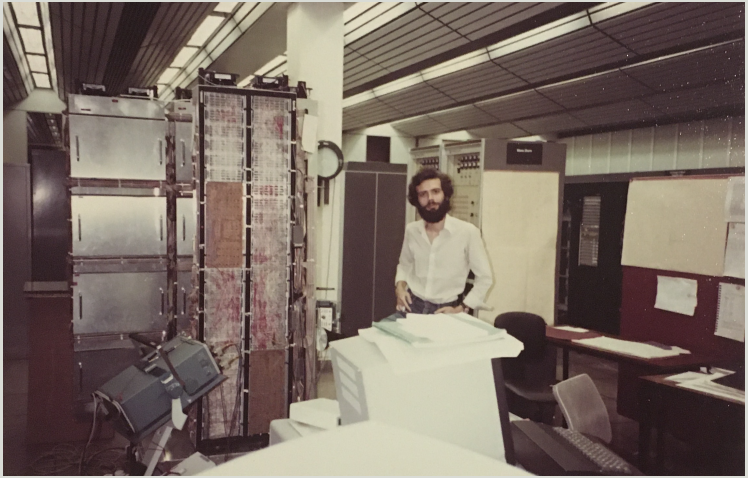


1980/81 — Manchester (MSc)

Pascal on Small Microcomputers

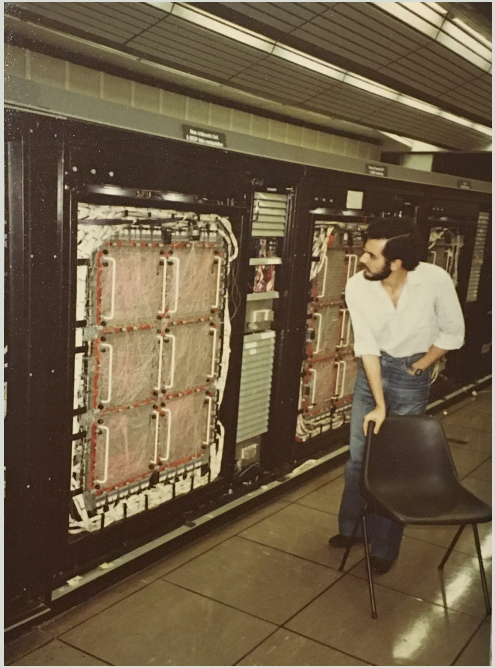


Manchester (CS)



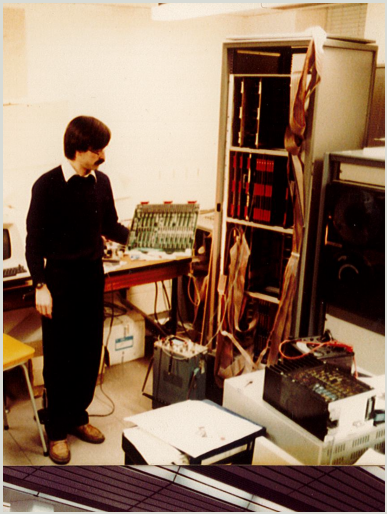
Manchester CS

O colega **Francisco Moura** na "sala das máquinas" do mesmo departamento.



1981 — The Manchester Dataflow Machine

(*In memoriam* José Guilherme Silva)



1981/84 — Manchester, PhD

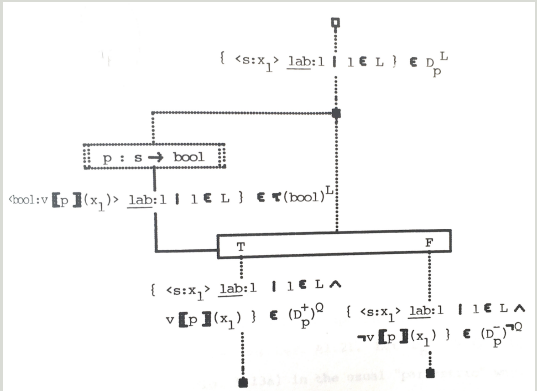


Fig. 5.14 - The Boolean switch.

THE FORMAL SEMANTICS OF DETERMINISTIC DATAFLOW PROGRAMS

by

J.N.F. Oliveira, M.Sc.

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Science

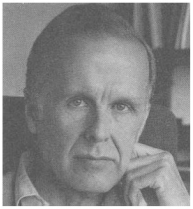
Department of Computer Science

February 1984

J. Backus (Turing Award 1978)

Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose



Conventional programming languages are growing ever more enormous, but not stronger. Inherent defects at the most basic level cause them to be both fat and weak: their primitive word-at-a-time style of programming inherited from their common ancestor—the von Neumann computer, their close coupling of semantics to state transitions, their division of programming into a world of expressions and a world of statements, their inability to effectively use powerful combining forms for building new programs from existing ones, and their lack of useful mathematical properties for reasoning about programs.

An alternative functional style of programming is

Revelação: Paralelismo = Paradigma funcional + Cálculo 

Impacto

Esta nova forma de ver a **computação** teve em mim um profundo impacto.

Impacto

Esta nova forma de ver a **computação** teve em mim um profundo impacto.

Os **programas** deixaram de ser meras sequências de instruções para serem, sobretudo, **redes** de fluxo de **informação.**

Impacto

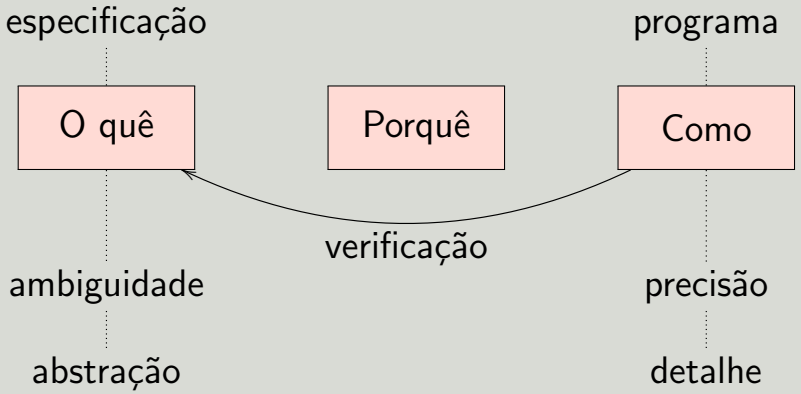
Esta nova forma de ver a **computação** teve em mim um profundo impacto.

Os **programas** deixaram de ser meras sequências de instruções para serem, sobretudo, **redes** de fluxo de **informação.**

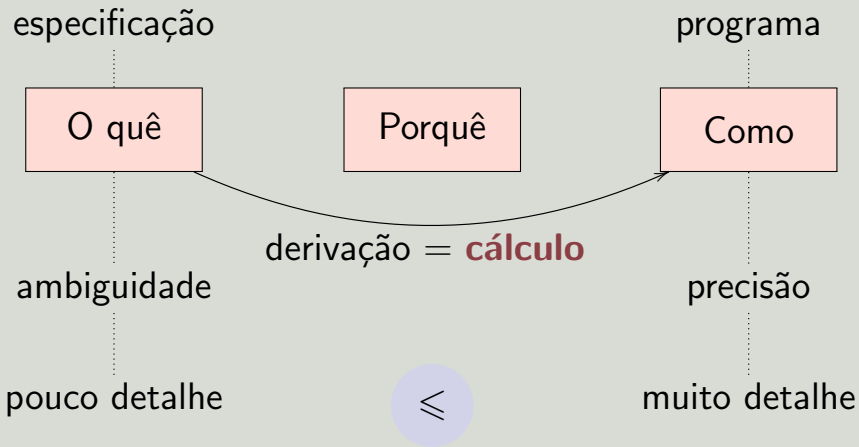
Foi a minha porta de entrada na programação **funcional** (e na **paralela**).

A programação imperativa ficou para trás como forma primordial de **arquitectar** programas.

Métodos formais — da verificação ao cálculo



Métodos formais



1992 — Refinement workshop (London)

Software Reification using the SETS Calculus

José N. Oliveira
DI/INESC, Universidade do Minho
Braga, Portugal

Abstract

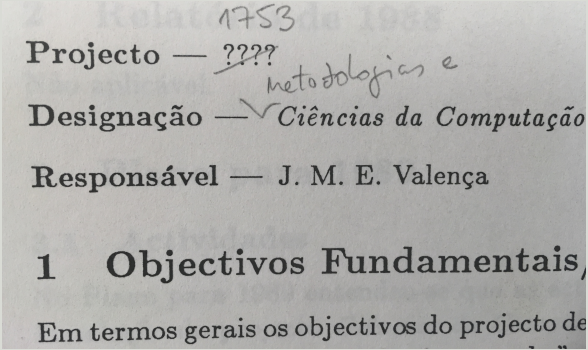
SETS is an emerging reification calculus for the derivation of implementations of *model-oriented* specifications of abstract data types.

This paper shows how *abstraction invariants* can be synthesized by calculation in SETS, and the potential of this calculus for assessing, comparing or classifying specifications.

The main results of the paper are concerned with a functorial approach to reification, particularly *wrt.* the systematic implementation of recursive data domains on non-recursive run-time environments. A final example of this class of implementations is provided.

Entretanto

1989 — **Protocolo UM/INESC**:


1753

Projecto — ????

metodologias e

Designação — *Ciências da Computação*

Responsável — J. M. E. Valença

1 Objectivos Fundamentais

Em termos gerais os objectivos do projecto de

Grupo 1753 — **Métodos Formais de Programação**.

Grupo Métodos Formais de Programação 18

Fernando Mário Martins

Grupo Métodos Formais de Programação 16

Francisco Soares Moura

Grupo Métodos Formais de Programação 17

Pedro Rangel Henriques

Grupo Métodos Formais de Programação 19

José João Almeida

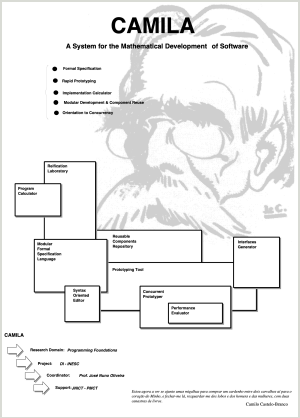
Grupo Métodos Formais de Programação 20

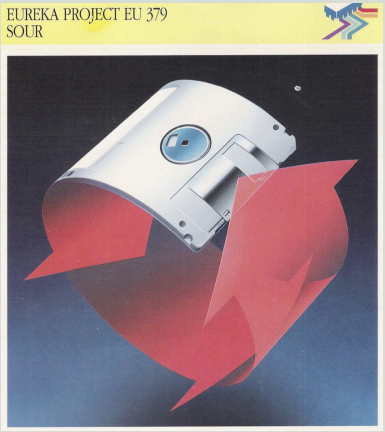
Luís Soares Barbosa

Grupo Métodos Formais de Programação 23

Rui Carlos Oliveira

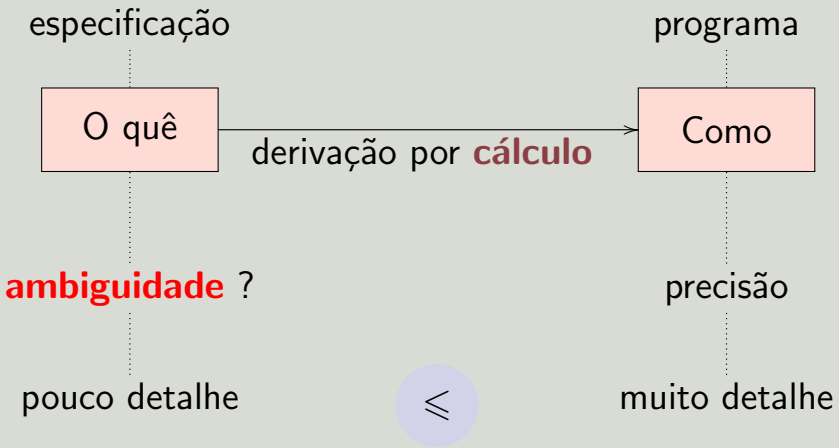
CAMILA + SOUR (1989-95)





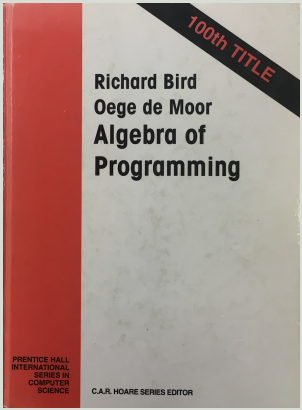
Tooling: ambiente para especificação e prototipagem funcional, etc etc

"Already there"? Não...



1997 — Algebra of Programming!

*[Programming is expressed by] "***formulæ** *and* **equations** *(...) which share the* **elegance** *of those which underlie* **physics** *and* **chemistry** *or any other branch of basic science".*



(Citação do Prefácio por C.A.R. Hoare)

Álgebra funcional (Backus etc)

Álgebra relacional (?)

Regresso às origens! (1860)

Augustus de Morgan (1806-71)

ON

THE SYLLOGISM, No. IV.

AND ON

THE LOGIC OF RELATIONS.

From the Transactions of the Cambridge Philosophical Society,
Vol. X. Part II.

By AUGUSTUS DE MORGAN, F.R.A.S. & C.P.S.
of trinity college, cambridge;
professor of mathematics in university college, london.

CAMBRIDGE:
PRINTED BY C. J. CLAY, M.A. AT THE UNIVERSITY PRESS.
M.DCCC.LX.

Frases "funcionais"

"Alice is the mother of the father of Bob"

$$Alice \xleftarrow{mother} x \xleftarrow{father} Bob$$

i.é

$$\begin{aligned} Alice &= mother\ (father\ (Bob)) \\ &= (mother \cdot father)\ (Bob) \end{aligned}$$

Recordar da matemática:

$$y = f\ (x)$$

Funções $\mapsto$ artigos **definidos** 

Frases "relacionais"

"Alice is an enemy of a son of Bob",

$$Alice \xleftarrow{enemy} x \xleftarrow{son} Bob$$

i.é

$$Alice\ (enemy \cdot son)\ Bob$$

Recordar da matemática:

$$y\ R\ x$$

Relações $\mapsto$ artigos **indefinidos** 

Voz passiva

Ao contrário das funções, as relações (R) admitem a **voz passiva** (R°), e.g.

Catarina **comeu** a laranja — $R = (comeu)$

A laranja **foi comida pela** Catarina — $R^\circ = (foi\ comida\ por)$.

i.e. $b\ R\ a \equiv a\ R^\circ\ b$

$$(R^\circ)^\circ = R$$
$$(R \cdot S)^\circ = S^\circ \cdot R^\circ$$

Voz passiva

Ao contrário das funções, as relações (R) admitem a **voz passiva** (R°), e.g.

Catarina **comeu** a laranja — $R = (comeu)$

A laranja **foi comida pela** Catarina — $R^\circ = (foi\ comida\ por)$.

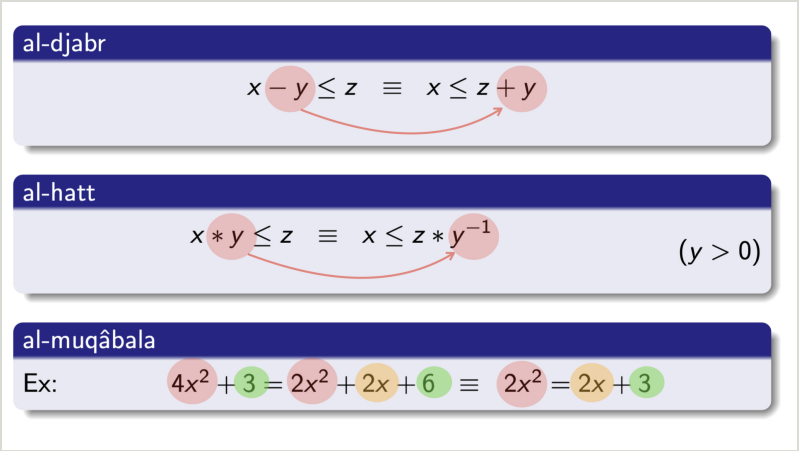
i.e. $$b\ R\ a \quad \equiv \quad a\ R^\circ\ b$$

Álgebra! 

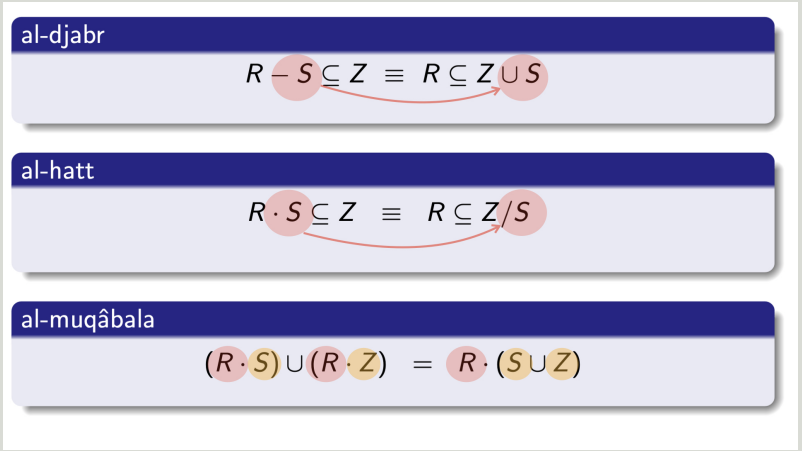
etc

$$(R^\circ)^\circ = R$$
$$(R \cdot S)^\circ = S^\circ \cdot R^\circ$$

Al-djabr? al-hatt? al-muqâbala?



Al-djabr; al-hatt; al-muqâbala



Roland Backhouse (1948-)

Mathematics of Program Construction

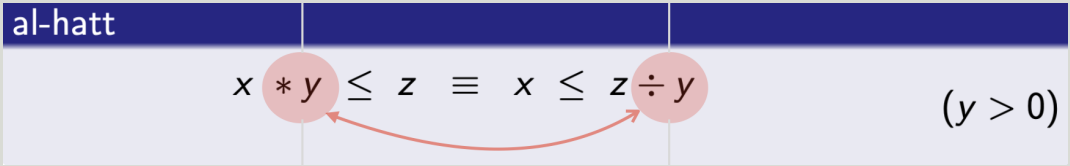
Draft

Roland Backhouse

with help from and contributions by
Marcel Bijsterveld, Henk Doornbos, Rik van Geldrop,
Diethard Michaelis, Jaap van der Woude

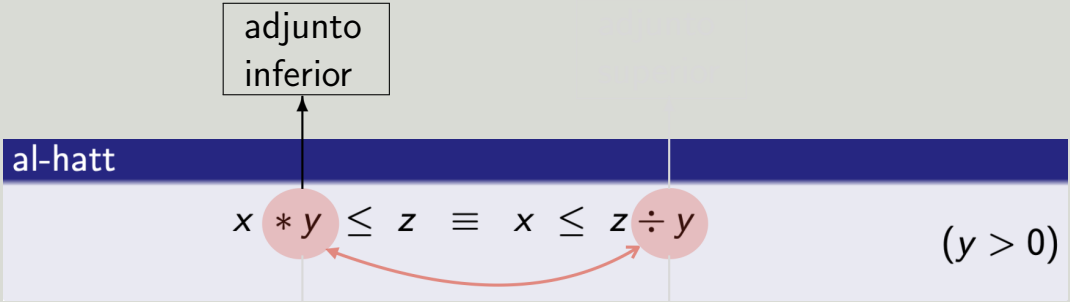
June 9, 2004

Conexões de Galois



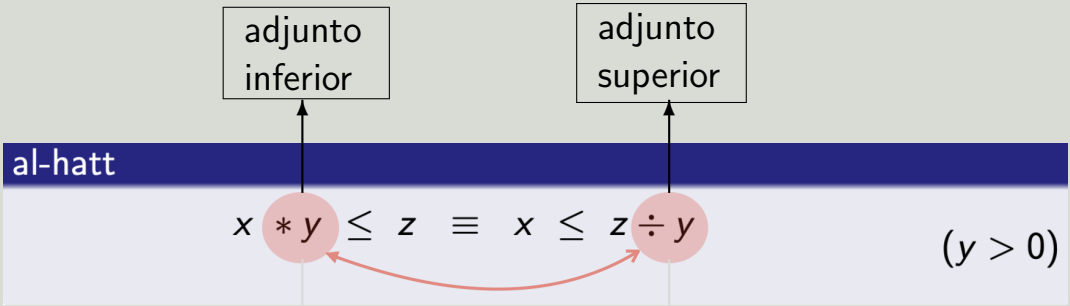
Al-hatt como conexão de Galois

Conexões de Galois



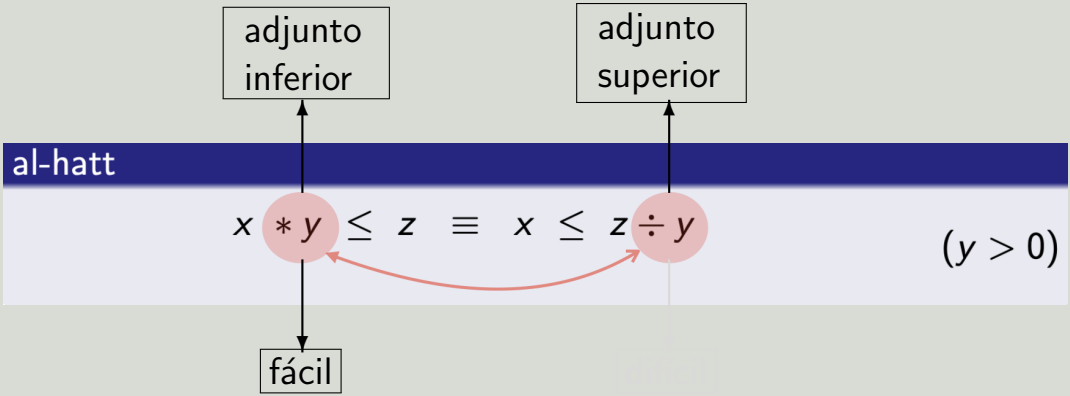
Al-hatt como conexão de Galois

Conexões de Galois



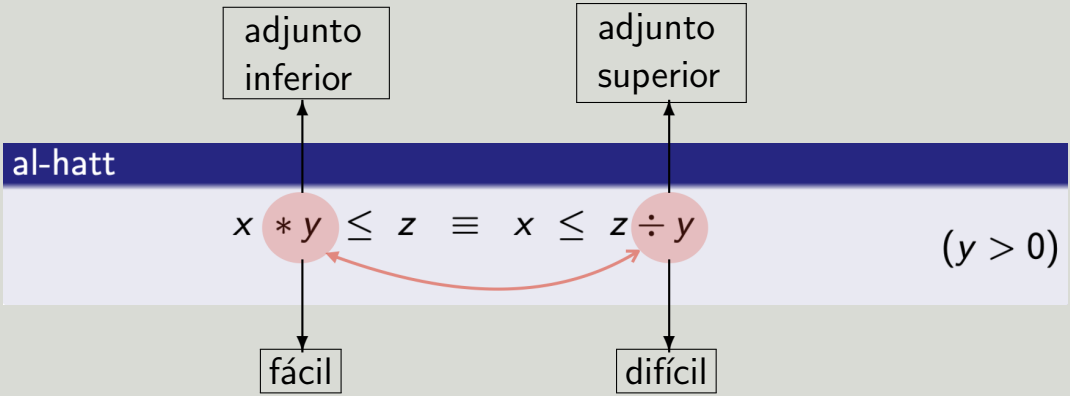
Al-hatt como conexão de Galois

Conexões de Galois



Al-hatt como conexão de Galois

Conexões de Galois



Al-hatt como conexão de Galois

al-hatt

$$x * y \leq z \equiv x \leq z \div y \qquad (y > 0)$$

$z \div y$

o maior x

z	y
$\cdots$	$z \div y$

al-hatt

$$x * y \leq z \equiv x \leq z \div y \qquad (y > 0)$$

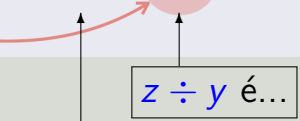
$z \div y$ é...

o maior x

z	y
$\cdots$	$z \div y$

al-hatt

$$x * y \leq z \equiv x \leq z \div y \qquad (y > 0)$$



z	y
$\cdots$	$z \div y$

al-hatt

$$x * y \leq z \equiv x \leq z \div y \qquad (y > 0)$$

que multi-plicado por y...

$z \div y$ é...

o maior x...

z	y
$\cdots$	$z \div y$

al-hatt

$$x * y \leq z \equiv x \leq z \div y \qquad (y > 0)$$

que multi-plicado por y...

não excede z.

o maior x...

$z \div y$ é...

z	y
$\cdots$	$z \div y$

Evariste Galois, 1831

"Certaines personnes ont [l'affectation] *d'éviter en apparence toute espèce de* **calcul**, *en traduisant par des phrases fort longues ce qui s'exprime très brèvement par* **l'algèbre**, *et ajoutant ainsi à la longueur des opérations, les longueurs d'un langage qui n'est pas fait pour les exprimer.*

Ces personnes-là sont en arrière de cent ans."

Evariste Galois (1811–1832)

2012 (WG2.1)

ELSEVIER

The Journal of Logic and Algebraic Programming

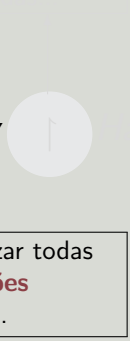
Volume 81, Issue 6, August 2012, Pages 680-704

Programming from Galois connections

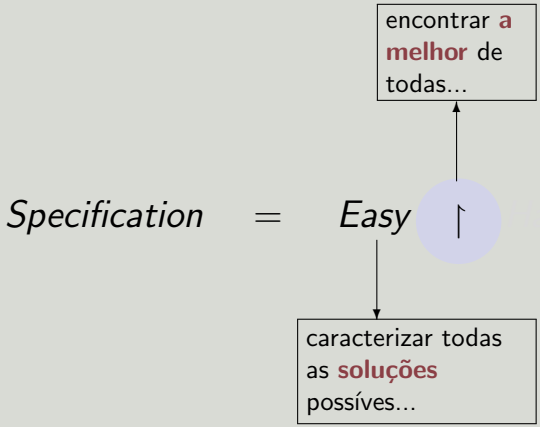
Shin-Cheng Mu [a], José Nuno Oliveira [b]

Os superlativos!

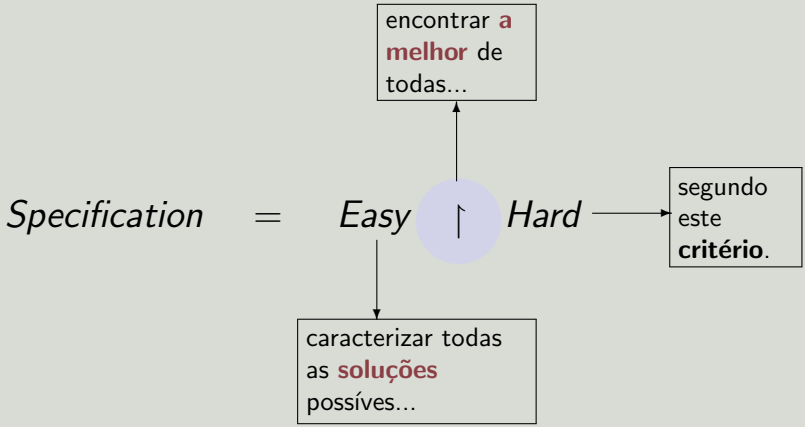
Specification $=$ *Easy*



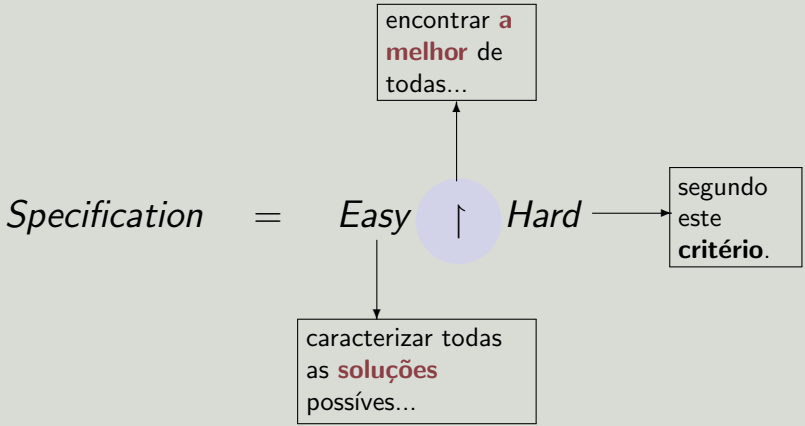
Os superlativos!



Os superlativos!



Os superlativos!



("al-hatt" : $X \subseteq R \wedge X \cdot R^{\circ} \subseteq S \equiv X \subseteq R \uparrow S$)

Os superlativos!

Não só em

---

best_sch t deverá dar **o melhor** (= **o mais rápido**) *escalonamento ("schedule") das tarefas em t conhecidos os respectivos tempos e as suas dependências.*

---

take n xs **o mais longo** *xs* *n*

Os superlativos!

Não só em

---

best_sch t deverá dar **o melhor** (= **o mais rápido**) *escalonamento ("schedule") das tarefas em t conhecidos os respectivos tempos e as suas dependências.*

---

mas também em

---

take n xs deverá dar **o mais longo** **prefixo** *de xs que não exceda n em* **comprimento**.

---

etc, etc, etc!

Comentário de um *referee*:

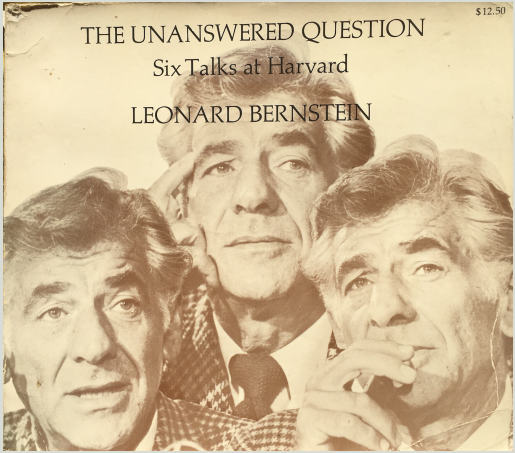
> *"— Exceptional Paper, with high perspectives. Starting with simple examples using Galois Connection author (sic) has been able to capture formally what one should consider as a way to build proved automatic programs from a specification."*

1976 — Bernstein at Harvard

"He immersed himself in Chomskyan **linguistics** *(...) so that he could then apply the principles of* **linguistics** *to* **music** *— thereby creating a brand new field of study.*

Ambitious? Oh, yes! Was he in over his head? Completely!"

(Jamie Bernstein).



Metáforas

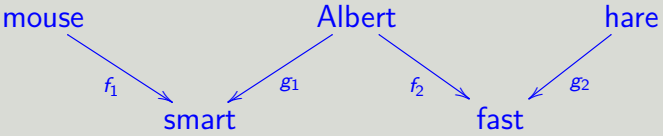
(a)	As Albert is as smart as a mouse and fast as a hare...	... he fled without a trace
	$\downarrow f$	
(b)	Albert ... smart ... mouse ... fast ... hare...	... fled without a trace
	$\downarrow g$	
(c)	Mousy 'Mr. Hare' ...	... fled without a trace

Chomsky: f e g transformam a "**deep** *structure*" (a) na "**surface** *structure*" (c).

O texto adquire uma dimensão **metafórica** à medida que se omitem palavras.

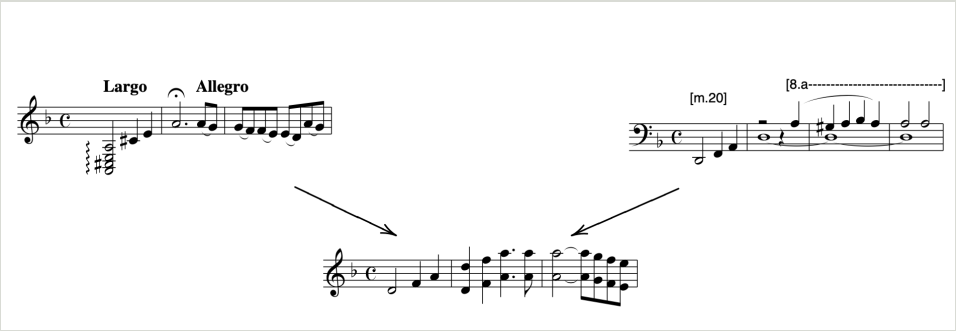
Como duas funções fazem uma metáfora

Justaposição de duas metáforas:



As funções em cada "V" são as **testemunhas** da respectiva metáfora.

Metáforas em música



(Beethoven, sonata opus 31 nº 2)

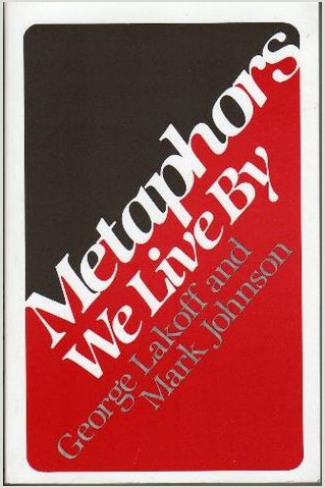
1980 — “Metaphors We Live By”

George **Lakoff** e Mark **Johnson** mostram como a nossa ”linguagem quotidiana” é metafórica, por exemplo:

- ’**Argument is War**’

  (e.g. “vencer um debate”, “conta-atacar nesse mesmo debate”, etc)

- **Time is Money**



1980 — "Metaphors We Live By"

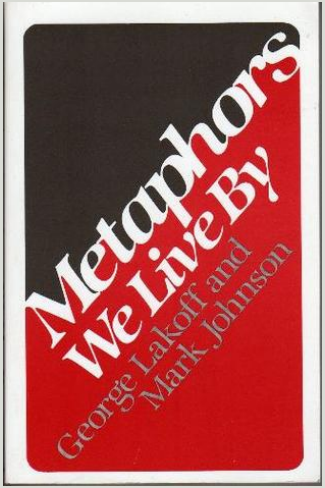
George **Lakoff** e Mark **Johnson** mostram como a nossa "linguagem quotidiana" é metafórica, por exemplo:

- '**Argument is War**'

  (e.g. "vencer um debate", "conta-atacar nesse mesmo debate", etc)

- '**Time is Money**'

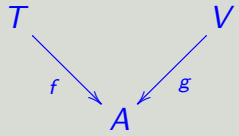
  (e.g. "gastar demasiado tempo numa tarefa", "investir tempo numa outra", etc)



1936 — Richards' Philosophy of Rhetoric

Segundo Richards, uma **metáfora** é sempre a interação (tensão) entre dois componentes, a saber:

- T (**tenor**) — o sujeito
- V é o **veículo** — a "imagem"
- A é um **atributo** partilhado



Pode-se interpretar uma metáfora como uma **relação binária**

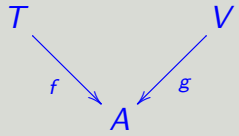
$$T(f^{\circ} \cdot g)V$$

onde $f^{\circ} \cdot g$ é a composição de f° com g e A

1936 — Richards' Philosophy of Rhetoric

Segundo Richards, uma **metáfora** é sempre a interação (tensão) entre dois componentes, a saber:

- T (**tenor**) — o sujeito
- V é o **veículo** — a "imagem"
- A é um **atributo** partilhado



Relacionalmente: uma metáfora é uma **relação binária**

$$T(f^{\circ} \cdot g)V$$

em que a **composição relacional** esconde o atributo (A).

Eça de Queirós? Mark Twain?

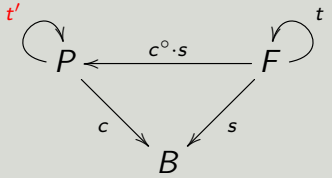
"Os políticos e as fraldas devem trocar-se frequentemente e pela mesma razão"

"Politicians and diapers should be changed often and for the same reason"

Eça de Queirós? Mark Twain?

"Os políticos e as fraldas devem trocar-se frequentemente e pela mesma razão"

Metáfora:



"Politicians and diapers should be changed often and for the same reason"

Axioma: $s\ (t\ x) = \mathrm{F}$ — induz *mudar* t' em P, etc etc.

Legenda: P = político (tenor); F = fralda (veículo); c = *corrompido* ; s = *suja* ; t = *mudar*.

Algebricidade

Aparentemente, um requisito "obstinadamente" reivindicado por **Saussure** nos seus escritos (ELG, p. 236): [2]

"(...) l'expression simple sera **algébrique** *ou ne sera pas"*

[2] S. Bouquet, *Ontologie et Épistemologie de la Linguistique dans les Textes Originaux de Ferdinand de Saussure*, U. Paris X, 2008, vol. XVIII, nr.3.

2013

On the *'A'* that links the *'M'*s of Maths, Music and Maps

J.N. Oliveira

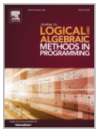
Dept. Informática,
Universidade do Minho
Braga, Portugal

21-23 November 2013
CEHUM Autumn Colloquium XV
Maths and Computer Science Panel
U.Minho, Braga

2018 — “Metaforismos”

Journal of Logical and Algebraic Methods in Programming

Volume 94, January 2018, Pages 15-44



Programming from metaphorisms

José Nuno Oliveira ✉

Metáforas em programação 

2018 — “Metaforismos”

Exemplo: o que é **ordenar** uma sequência?

$$
\begin{aligned}
&Sort = Easy \mid Hard \\
&\quad \textbf{where} \\
&\qquad Easy = bag^{\circ} \cdot bag \\
&\qquad Hard = ordered^{\circ} \cdot true
\end{aligned}
$$

$bag\ (\times)$ x

2018 — “Metaforismos”

Exemplo: o que é **ordenar** uma sequência?

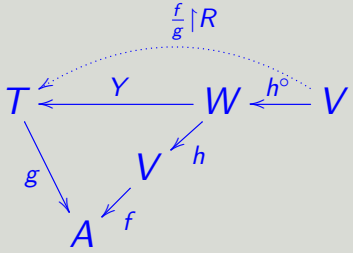
$$
\begin{aligned}
&Sort = Easy \upharpoonright Hard \\
&\quad \textbf{where} \\
&\qquad Easy = bag^{\circ} \cdot bag && \text{— metáfora "invariante"} \\
&\qquad Hard = ordered^{\circ} \cdot true && \text{— metáfora "variante"}
\end{aligned}
$$

Legenda: $bag\ (x)$ = elementos da sequência x, inc. quantas vezes se repetem.

2018 — “Metaforismos”

Onde investir?

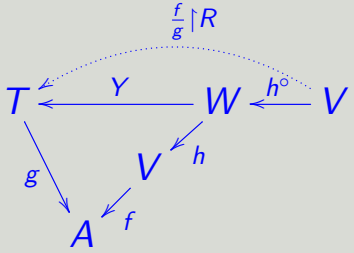
No **tenor**? $\mapsto$ *mergesort* 



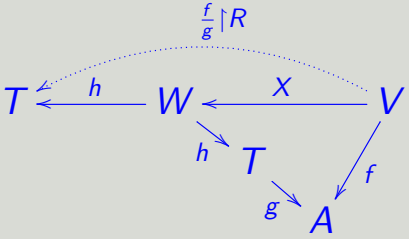
2018 — “Metaforismos”

Onde investir?

No **tenor**? $\mapsto$ *mergesort* 



No **veículo**? $\mapsto$ *quicksort* 



Mas nem só de metáforas vive o Homem...

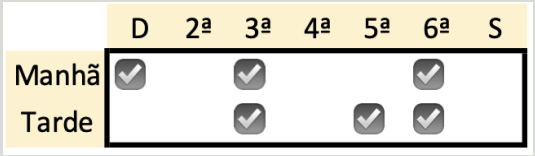
- Data-driven?
- Data-enabled?
- Frequentista probabilística?
- Frequentista qualitativa?
- Dados massivos?

Mas nem só de metáforas vive o Homem...

- **Data-science**?
- **Data-mining**?
- Programação **probabilística**?
- Programação **quântica**?
- **Redes neuronais**?

Mas nem só de metáforas vive o Homem...

No princípio, bastava saber-se se ia chover ou não...



isto é, **1** (chove), **0** (não chove):

	D	2ª	3ª	4ª	5ª	6ª	S
Manhã	1	0	1	0	0	1	0
Tarde	0	0	1	0	1	1	0

Mas nem só de metáforas vive o Homem...

Agora somos mais exigentes: queremos saber a **probabilidade** de chover etc etc.

	D	2ª	3ª	4ª	5ª	6ª	S
Manhã	1	0	0.3	0	0	0.9	0
Tarde	0	0	0.7	0	1	0.1	0

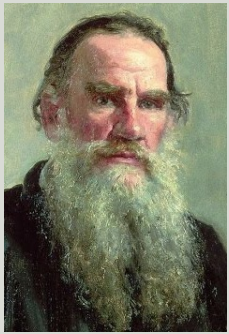
Queremos **medir** tudo — **prever** tudo — explicar tudo por **números**.

Data-mining, **Data-science**!

1869

"Only by taking infinitesimally small units for observation (the **differential** *of history, that is, the individual tendencies of men) and attaining to the art of* **integrating** *them (that is, finding the sum of these infinitesimals) can we hope to arrive at the laws of history."*

Leo Tolstoy, "War and Peace"
- Book XI, Chap.II (1869)



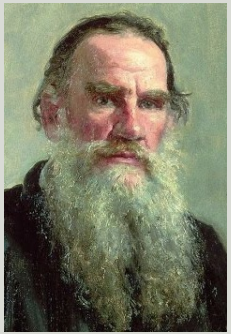
L. Tolstoy (1828–1910)

150+ anos depois, não é isto que tentamos fazer em **data-science**, **data-mining** etc?

1869

"Only by taking infinitesimally small units for observation (the **differential** *of history, that is, the individual tendencies of men) and attaining to the art of* **integrating** *them (that is, finding the sum of these infinitesimals) can we hope to arrive at the laws of history."*

Leo Tolstoy, "War and Peace"
- Book XI, Chap.II (1869)



L. Tolstoy (1828–1910)

150+ anos depois, não é isto que tentamos fazer em **data-science**, **data-mining** etc?

Como se domina essa "**art of integration**"?

Relações? Matrizes! Vectores!

Relação binária (álgebra **relacional**):

	D	2ª	3ª	4ª	5ª	6ª	S
Manhã	1	0	1	0	0	1	0
Tarde	0	0	1	0	1	1	0

Matriz (álgebra **linear**):

	D	2ª	3ª	4ª	5ª	6ª	S
Manhã	1	0	0.3	0	0	0.9	0
Tarde	0	0	0.7	0	1	0.1	0

Relações "=" Matrizes

Relações binárias e matrizes são *abstractamente* a mesma coisa 

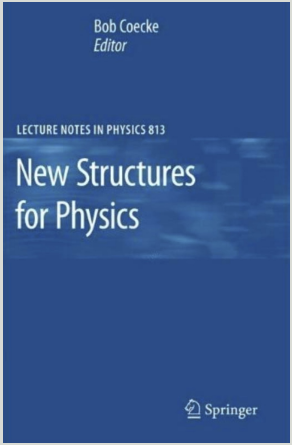
Uma **matriz** mais não é que uma **relação** binária **quantitativa**.

Relações e matrizes partilham muita teoria em comum.

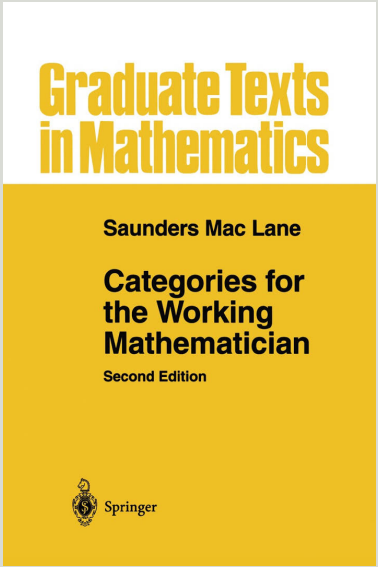
Como exprimi-lo?

2011 — Computação quântica

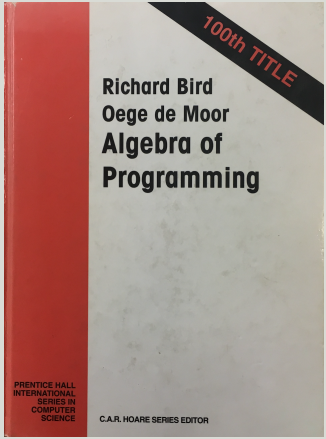
"(...) **Rel** *[the category of relations] possesses more 'quantum features' than the category Set of sets and functions [...] The categories* **FdHilb** *and* **Rel** *moreover admit a categorical* **matrix calculus***."*

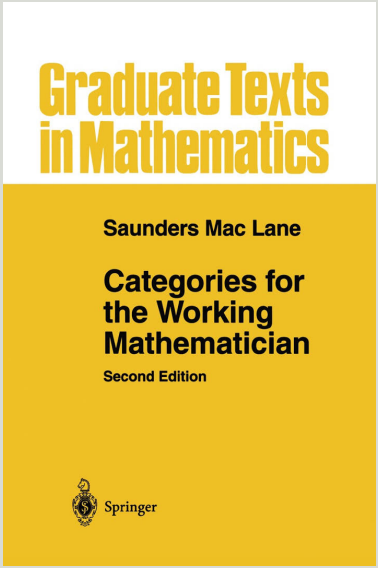


“Você disse... categoria?”



“Você disse... categoria?”





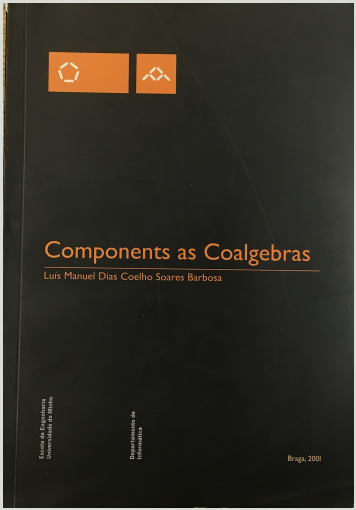
Categorias “in house”

Painless Implementation of Algebraic Specifications or How to Live Happily with Categories

José Manuel E. Valença *

Abstract

In this work we seek to show that formal methods of programming can be taught at the introductory level of a computer science degree using the language *ML* as support. We will illustrate our experience with two case studies: implementation of homogeneous algebras and the implementation of parameterized theories.



2013 — Visão categorial da Algebra Linear

ELSEVIER

Science of Computer Programming

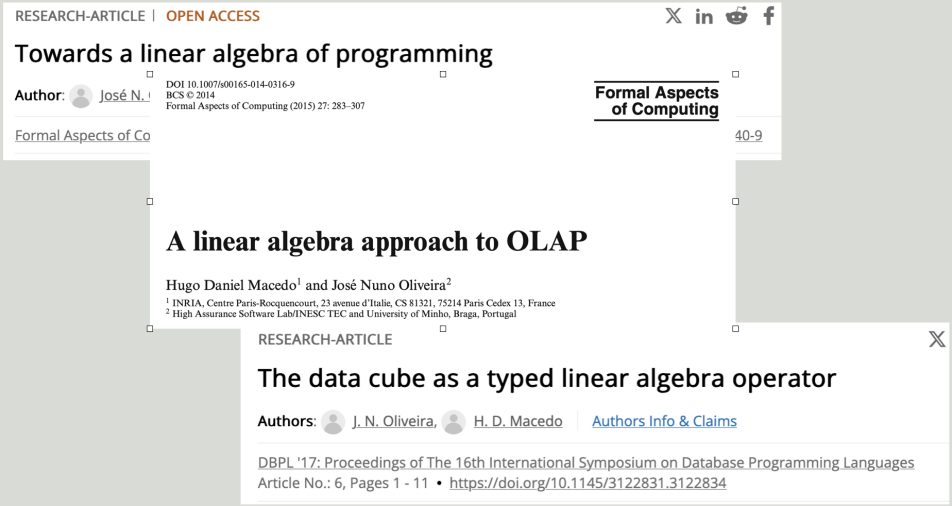
Volume 78, Issue 11, 1 November 2013, Pages 2160-2191

Typing linear algebra: A biproduct-oriented approach

Hugo Daniel Macedo, José Nuno Oliveira

Show more

LAoP — “Linear Algebra of Programming”



RESEARCH-ARTICLE | OPEN ACCESS

Towards a linear algebra of programming

Author: José N. [illegible]

Formal Aspects of Co [illegible] 40-9

DOI 10.1007/s00165-014-0316-9
BCS © 2014
Formal Aspects of Computing (2015) 27: 283–307

Formal Aspects of Computing

A linear algebra approach to OLAP

Hugo Daniel Macedo[1] and José Nuno Oliveira[2]

[1] INRIA, Centre Paris-Rocquencourt, 23 avenue d'Italie, CS 81321, 75214 Paris Cedex 13, France
[2] High Assurance Software Lab/INESC TEC and University of Minho, Braga, Portugal

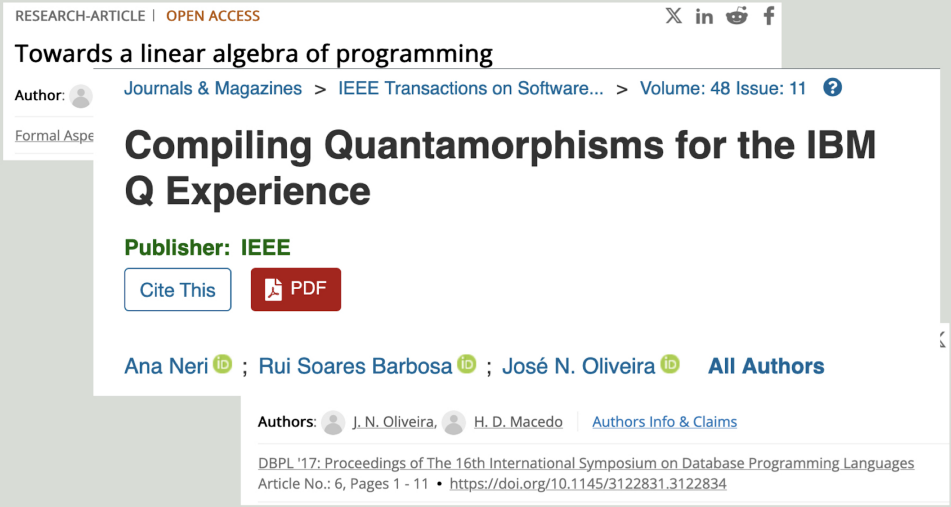
RESEARCH-ARTICLE

The data cube as a typed linear algebra operator

Authors: J. N. Oliveira, H. D. Macedo Authors Info & Claims

DBPL '17: Proceedings of The 16th International Symposium on Database Programming Languages
Article No.: 6, Pages 1 - 11 • https://doi.org/10.1145/3122831.3122834

Aplicação à computação quântica

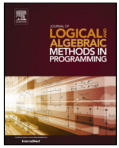


Um lema simples

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp



“Keep definition, change category” — A practical approach to state-based system calculi

José Nuno Oliveira [a,*], Victor Cacciari Miraldo [b,1]

[a] *High Assurance Software Laboratory, INESC TEC and University of Minho, Braga, Portugal*
[b] *Dep. of Information and Computing Sciences, Universiteit Utrecht, Utrecht, Netherlands*

O valor da simplicidade

"Simplicity does not precede complexity, but follows it."

(Edsger Dijkstra)

E.W. Dijkstra (1920-2002)

2025 — Ensino

2025 — Ensino

Tempos de dramáticas mudanças.

Sabemos que não podemos continuar a *"ir por aí"*... mas tardamos a reagir.

Modelo (secular) universitário em estado de choque.

Há quem há muito não esteja nada feliz...

Vinton Cerf (1943-)

ACM President

DOI:10.1145/2347736.2347737

Where is the Science in Computer Science?

"... we have a responsibility to pursue the **science** *in computer science. We must develop better tools and much deeper understanding of the systems we invent and a far greater ability to make* **predictions** *about the behavior of these complex, connected, and interacting systems.".*

(V.G. Cerf, Letter from the ACM President, CACM 55(10), Oct. 2012)

2025 — Ensino

Formação **insuficiente** face aos novos desafios:

- **Computação quântica**
- **Redes neuronais**
- **Software confiável**

2025 — Ensino

Formação **insuficiente** face aos novos desafios:

- **Computação quântica**
- **Redes neuronais**
- **Software confiável**

Como dizia Amílcar Sernadas, a **Álgebra Linear** é cada vez mais a

> *"língua franca da* **Computação***".*

Mas ... em que **espaço lectivo**?

A. Sernadas (1952–2017)

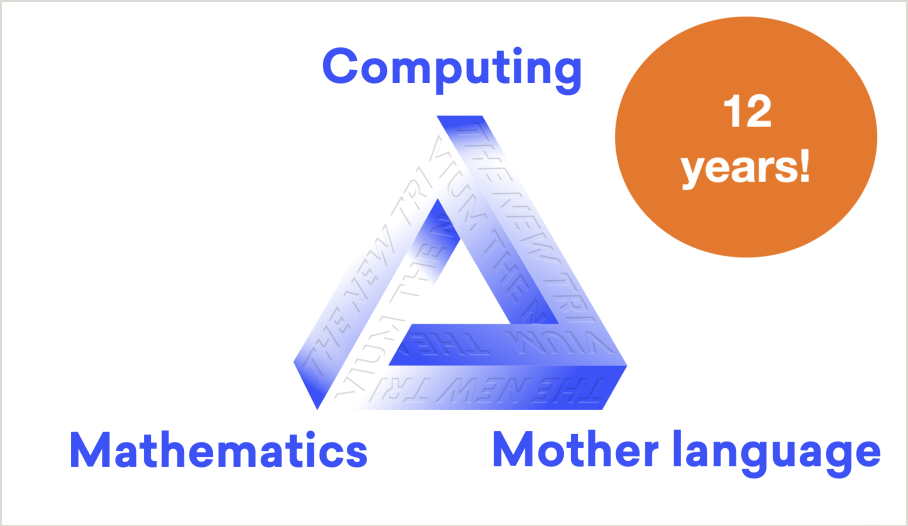
Carl Sagan

"We live in a society exquisitely dependent on science and technology, in which hardly anyone knows anything about science and technology. (...) If we continue to accumulate only power and not wisdom, we will surely destroy ourselves".

Carl Sagan (1934–1996)

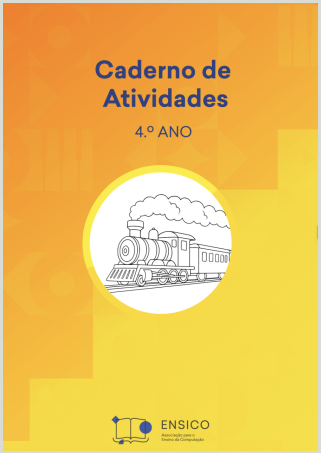
Ensino de Computação nas escolas

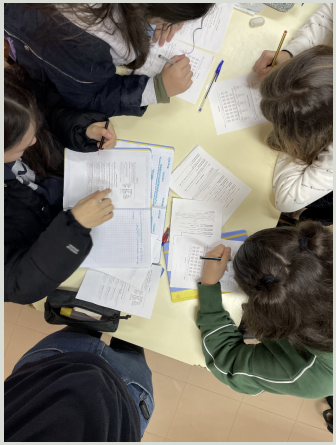
ENSICO

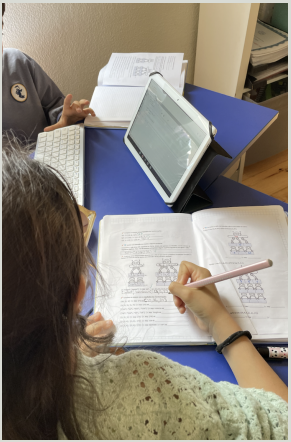


Ensino de Computação nas escolas

ENSICO







Aritmética binária

Mas, como é que ele soma os próprios bits?

Por exemplo,

1 + 1 + 1 = 1 1

como é calculado?

				1	1				
	56			1	1	1	0	0	0
+	25		+	0	1	1	0	0	1
	81		1	0	1	0	0	0	1

Como há 40 anos...

Lutemos por um ensino **digno** da **Computação** ao nível **K12**.

Como há 40 anos José Valença e Amílcar Sernadas o fizeram, a nível **universitário**.

A **Computação** entrou pelas nossas vidas, está a condicioná-las, e não é lugar para **amadorismos**.

Ensino — o futuro

"A literacia digital ao nível **K12** (...) abrirá espaço para o ensino de **matérias mais ambiciosas** ao nível do **ensino superior**.

Tem, assim, potencial para vir a **revolucionar** e atualizar os **conteúdos pedagógicos** dos cursos de informática e computação, gerando uma dinâmica com um potencial económico valioso num país que (...) tem como principal matéria-prima a massa cinzenta e a energia dos seus cidadãos."[3]

[3] In *'De pequenino se torce o pepino'*, BIP Inesctec Magazine, 2021-10-01.

"Quien sabe por Algebra, sabe scientificamente"

(Pedro Nunes, *Libro de Algebra en Arithmetica y Geometria*, 1567, fol. 270)

Obrigado