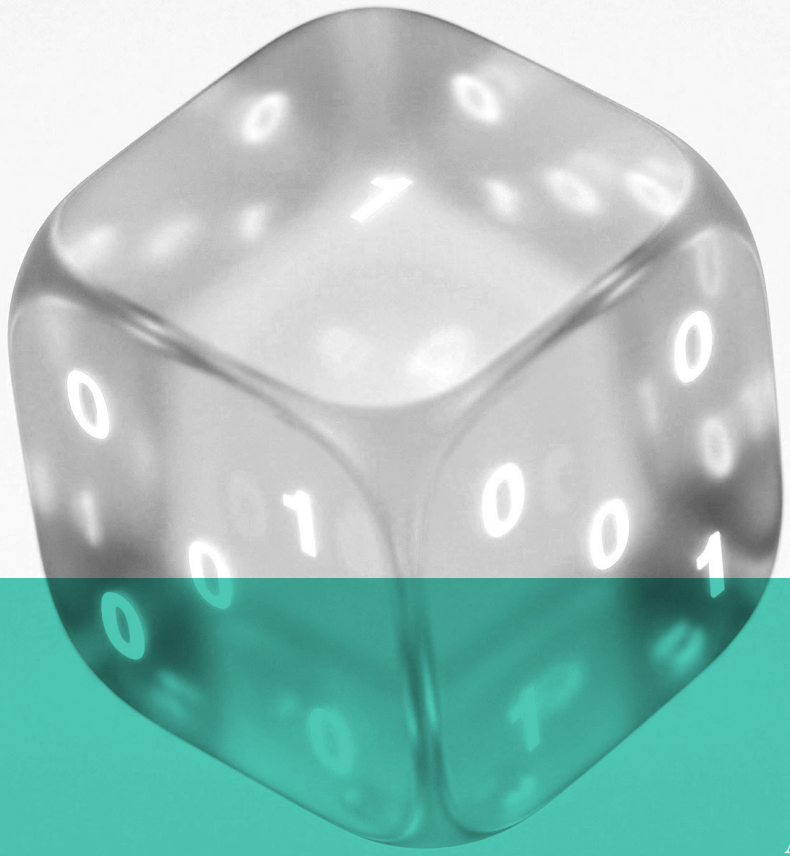
A máquina que só podía sumar xa é quen de argallar

Como funcionan por dentro as intelixencias artificiais

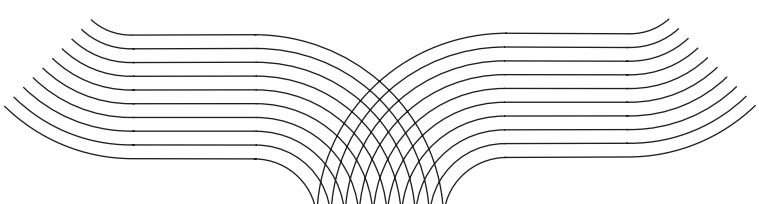


Autores:

Iván Area Carracedo
Manuel Caeiro Rodríguez
Arno Formella

Miscelánea
Serie de textos misceláneos

Servizo de Publicacións
Universidade Vigo



Miscelánea
Serie de textos misceláneos

Edición
Universidade de Vigo
Servizo de Publicacións
Rúa de Leonardo da Vinci, s/n
36310 Vigo

Revisión lingüística
Alejandra Freire Amorín

Deseño gráfico
Tania Sueiro Graña
Miranda Rodríguez Martínez
Área de Imaxe

Imaxe da portada
Adobe Stock

Maquetación interior
Os autores

ISBN
978-84-1188-132-6

Depósito legal
VG 662-2026

DOI
doi.org/10.35869/1188.1326

© Servizo de Publicacións da Universidade de Vigo, 2026
© Iván Area Carracedo, Manuel Caeiro Rodríguez e Arno Formella

Ao ser esta editorial membro da une, garántense a difusión e a comercialización das súas publicacións no ámbito nacional e internacional.

Servizo de Publicacións
Universidade de Vigo

HR EXCELLENCE IN RESEARCH

Este volume publícase
co financiamento da

XUNTA
DE GALICIA



This license enables reusers to distribute (i.e. download, copy and share) the material in any medium or format for noncommercial purposes only if attribution is given to the creator. CC BY-NC-ND 4.0 includes the following elements:

BY: Credit must be given to the creator.

NC: Only noncommercial use of this work is permitted.
Noncommercial means not primarily intended for or directed towards commercial advantage or monetary compensation.

ND: No derivatives or adaptations of this work are permitted.

A máquina que só podía sumar xa é quen de argallar

Como funcionan por dentro as intelixencias artificiais

Autores:

Iván Area Carracedo
Manuel Caeiro Rodríguez
Arno Formella

Índice xeral

Proemio ix

Como ler este libro xxix

BLOQUE 0 A pregunta que ninguén fai

01 Como o fai a túa calculadora? 3

1.1 Unha pregunta que parece parva 3
1.2 Multiplicar sen saber a táboa 6
1.3 A raíz cadrada: estimar e corrixir 11
1.4 O seno: xirar no canto de medir 14
1.5 A calculadora aproxima e non o di 17
1.6 Tres leccións que valen para todo o libro 18

02 Que quere dicir «saber como funciona»? 21

2.1 A ilusión de entender 21

2.2 Tres preguntas que non son a mesma 23
2.3 Que é exactamente un algoritmo 25
2.4 Dentro non hai o que hai fóra 29
2.5 ELIZA e o xogo da imitación 32
2.6 Que imos entender e que non 35

BLOQUE I Todo é número

03 Contar con dous dedos 41

3.1 Dez dedos, dez cifras 41
3.2 Contar cando só hai dúas opcións 43
3.3 Por que os aparellos escolleron o dous 48
3.4 Sumar en binario: unha táboa de catro liñas 51
3.5 E os números negativos? 54
3.6 E os números con coma? 55

04 Letras, cores e cancións 59

4.1 Un número para cada letra 59
4.2 O galego dentro da máquina 62
4.3 O son é unha lista de alturas 66
4.4 Unha imaxe é unha táboa de números 68
4.5 Comprimir é tirar o que non se nota 70
4.6 O que se perde ao pasar de analóxico a dixital 72

05 Da corrente á decisión 77

5.1 Verdadeiro e falso tamén son dous 77

5.2 Tres portas: e, ou, non 78
5.3 Sumar con portas 83
5.4 De sumar a decidir 86
5.5 O programa tamén é un número 90
5.6 Por que todo isto é tan rápido 92

BLOQUE II A medición da linguaxe

06 Dime con quen andas 99

6.1 Antes das palabras escritas 99
6.2 Un número por palabra non serve 102
6.3 Un dicionario tampouco resolve nada 104
6.4 A palabra coñécese polas súas compañías 105
6.5 Contar veciñas 106
6.6 O que esta idea captura e o que non 110
6.7 As palabras que significan dúas cousas 111

07 O mapa das palabras 115

7.1 De dúas coordenadas a moitas 115
7.2 Que é un vector, sen medo 118
7.3 Medir a distancia entre dúas palabras 122
7.4 Direccións que significan algo 128
7.5 Un mapa que ninguén debuxou 130
7.6 O mapa herda o que hai no texto 133

08 Cortar as palabras en anacos 137

8.1 Por que non vale «unha palabra, un número» 137
8.2 A solución de compromiso 139
8.3 Como se decide onde cortar 141
8.4 O galego sae peor parado 146
8.5 Consecuencias que se ven 148
8.6 Cantos anacos caben 151

BLOQUE III A aprendizaxe da linguaxe

09 Adiviñar a palabra seguinte 157

9.1 O xogo de Shannon 159
9.2 Contar continuacións 163
9.3 Repartir cento por cento 166
9.4 Por que contar non abonda 168
9.5 Predicir obriga a comprimir 169
9.6 Da palabra seguinte ao texto enteiro 173

10 Que é «aprender» para unha máquina 179

10.1 Aprender é reducir un erro 180
10.2 Os mandos que se poden axustar 182
10.3 O erro: medir canto se desatina 185
10.4 Unha fórmula para mover dous mandos 188
10.5 Outra forma de mover os mandos: a ladeira na néboa 191
10.6 O tamaño do paso 197
10.7 Millóns de mandos á vez 199
10.8 Que é aprender, na práctica 201

11 Neuronas que non son neuronas 205

11.1 Unha suma con pesos e unha función de activación 206
11.2 O perceptrón e o seu límite 208
11.3 Poñer capas 212
11.4 Que aprende cada capa 215
11.5 En que se parece a un cerebro e en que non 216
11.6 Por que fixo falta esperar setenta anos 217

BLOQUE IV A viaxe pola linguaxe

12 Mirar cara atrás: a atención 227

12.1 Palabras que dependen doutras 228
12.2 Un exemplo que se resolve mirando atrás 230
12.3 Pregunta, chave e valor 231
12.4 Repartir a atención 233
12.5 Varias miradas á vez 240
12.6 Quen decide que preguntas, que chaves e que valores? 242
12.7 Por que esta peza cambiou todo 244
12.8 En que se diferencian uns modelos doutros 248

13 Da túa pregunta á súa resposta 253

13.1 O percorrido completo 254
13.2 A lista de candidatos 256
13.3 Escoller 257
13.4 A temperatura, explicada de verdade 259
13.5 Palabra a palabra 261

13.6 E cando para? 265
13.7 Por que *esquece* 268

14 Por que inventa contos e feitos 273

14.1 Dentro non hai ningunha comprobación 273
14.2 O plausíbel e o verdadeiro 275
14.3 Por que as citas falsas son tan verosímiles 279
14.4 Por que soa tan seguro 281
14.5 O que si axuda 283
14.6 Cando fiarse e cando non 284

BLOQUE V O que non se ve da máquina

15 O que custa 291

15.1 De onde saíu o texto 292
15.2 As persoas que non se ven 295
15.3 Enerxía e auga 298
15.4 Quen pode pagalo 303
15.5 Os nesgos non son unha avaría 304
15.6 O que se pode esixir 307

16 O que segue sendo noso 311

16.1 Producir e responder non é o mesmo 313
16.2 Ler criticamente unha resposta 313
16.3 O criterio non se automatiza 316
16.4 Escribir a man, pensar amodo 317

16.5 Que ensinar agora 320
16.6 A pregunta do principio 322

Epílogo: onde poñer a panca 327

Apéndices

A As receitas da calculadora, por dentro 335

A.1 Onde remata o circuíto e empeza a receita 335
A.2 Canto custa multiplicar de verdade 337
A.3 A raíz cadrada que se aprendía na escola 339
A.4 A raíz cadrada que usa a máquina 344
A.5 E o seno? Xirar no canto de medir 351

B Números con coma dentro da máquina 355

B.1 Números con coma 355
B.2 Por que 0,1 non existe dentro 360

C Respostas do bloque 0 365

D Respostas do bloque I 369

E Respostas do bloque II 375

F
G
H
I
J

Respostas do bloque III 383

Respostas do bloque IV 393

Respostas do bloque V 401

Respostas dos apéndices A e B 407

Glosario 411

Bibliografía 415

Aprender 427

Argallar 429

Proemio

Este libro naceu dunha pregunta que o primeiro autor fixo nunha aula. Unha pregunta que non tivo resposta.

A pregunta non estaba prevista. Xurdiu falando sobre o funcionamento da calculadora do teléfono móbil: que fai por dentro cando lle pides a raíz cadrada de dous? Houbo un silencio breve, un par de respostas prudentes e despois algo máis interesante có silencio: varias persoas dixeron, con moita naturalidade, que nunca se lles pasara pola cabeza preguntarse tal cousa.

Unha aclaración, xa que vén a conto. Este libro escríbeno tres persoas e vai en plural porque case todo o pensamos entre os tres. Cando algo lle pasou a unha soa, como esta aula, dise de quen foi: *o primeiro autor* é Iván Area, *o segundo* é Manuel Caeiro e *o terceiro* é Arno Formella. Non hai máis misterio e non volveremos avisar.

Volvamos á aula. O primeiro autor saíu de alí coa pregunta na cabeza e contouno. Démoslle voltas e quedamos os tres coa mesma dúbida. Non sabemos que nos preocupou máis: que non o souberan ou que nunca se preguntasen por iso. Inclinámonos polo segundo. Non saber é o estado natural de calquera de nós diante da maior parte das cousas e arránxase. Non ter curiosidade por saber é máis difícil de arranxar, porque non se nota. Quen non sabe algo pode buscalo; quen nin sequera formulou a pregunta non ten nada que buscar.

A escola e a universidade miden as respostas, non as preguntas. Un exame detecta que quen estuda non sabe algo, pero non se algunha vez se preguntou por iso. Así que o problema máis grave, a falta de curiosidade, é xusto o que o sistema non ve. E nun mundo cheo de respostas, que vale máis: saber responder ou saber preguntar?

Ese mesmo mes chegounos outra consulta: alguén que, sen coñecementos de

matemáticas nin de tecnoloxía, buscaba un libro que lle explicase como funcionan as intelixencias artificiais xenerativas. Puxémonos a buscalo. Atopamos libros excelentes para quen xa sabe programar, ensaios moi bos sobre as consecuencias ou os perigos da tecnoloxía e manuais de trucos para escribir instrucións que non explican nada. O que buscaba a persoa que preguntaba estaría no medio de todo isto, pero non o atopamos.

Ante unha pregunta chea de curiosidade pero sen resposta, decidimos escribir ese libro nós mesmos. Empezando, iso si, pola calculadora.

«Que como funciona? Non o sei nin me importa»

O caso da calculadora non é excepcional. É o caso normal. Somos ignorantes e non por preguiza: ignoramos, sobre todo, aquilo que non nos dá nin pracer, nin dor, nin sorpresa. A calculadora non dá ningunha das tres cousas. Funciona e xa está. (Hai quen si sente pracer cunha demostración matemática ben feita: existe un libro enteiro dedicado a iso, *Proofs from THE BOOK* [1]. Os números entre corchetes, coma este, remiten á bibliografía do final do libro.)

Algúns exemplos: usas o ascensor todos os días e case seguro que non sabes como funciona. Hai cousas que só pasan nas películas: unhas persoas dentro dun ascensor, un cable que rompe e a cabina que se precipita ao baleiro. No mundo real é practicamente imposíbel que caia, porque hai un freo que actúa *precisamente* cando se rompe o cable, non cando funciona. É o invento de Elisha Otis, de mediados do século XIX e é precioso. Outro: abres a neveira e non sabes por que dentro fai frío mentres a parte de atrás quenta. Parecido: tiras da cisterna sen ter visto nunca o sifón que impide que suba o cheiro. E, sobre todo, escribes tres palabras nun buscador, recibes en dúas décimas de segundo unha lista ordenada de entre miles de millóns de páxinas e non che parece raro.

UN POUCO DE HISTORIA *o xeado que lle deu nome a un fenómeno*

Alguén che di que, se queres conxelar un xeado máis rápido, primeiro tes que quentar a mestura. Vaia parvada, non?

En 1963, nunha escola de Tanganica (hoxe Tanzania), un rapaz de trece anos chamado Erasto B. Mpemba tiña présa e meteu no conxelador a mestura do xeado sen deixala arrefriar. Conxelouse antes cá do resto da clase. Cando o contou, riron del. El seguiu a preguntar e, seis anos despois, publicou o resultado co físico Denis Osborne nunha revista de ensino da física, nun artigo titulado, con retranca, Cool? [102].

O fenómeno chámase desde entón ***efecto Mpemba*** *e, o que é máis curioso, segue sen estar resolto: hai experimentos que o atopan, outros que non e discusións sobre que significa exactamente «conxelarse». Podería acabar en nada.*

O que non é nada é a historia. Un rapaz de trece anos fixo unha pregunta que todo o mundo consideraba parva e acabou nos libros, co seu propio apelido detrás: efecto Mpemba. Ese é exactamente o valor que lle damos aquí a preguntar como funcionan as cousas.

EN PALABRAS SINXELAS *cando non queremos saber?*

Non é preguiza nin falta de capacidade. É un acordo tácito e bastante razoábel: a vida non dá para entender todo e as cousas ben feitas están deseñadas precisamente para que non teñas que pensar nelas. Un interruptor bo é o que non che obriga nin a pensar en como accionalo, xa que a forma do trebello suxire o que hai que facer con el. En inglés chámanlle a isto *affordance*: a forma do trebello convida a usalo dun xeito determinado, sen instrucións. Con matices, claro: un interruptor tampouco é evidente para quen nunca viu ningún. O que hoxe parece natural aprendeuse nalgún momento.

Saber que un ascensor non pode caer, ou como facer os xeados máis rápido, non parecen cuestións importantes. Pero o problema da ignorancia é que é moi atrevida. Ese acordo tácito que anula a curiosidade serve de marabilla para o interruptor, pero pode levarnos a situacións comprometidas cando se aplica tamén a unha máquina que escribe informes, corrixe exames, filtra currículos e responde preguntas médicas. Aí, non entender xa non é cómodo: é arriscado.

MÍRAO *canto cabe nunha vida?*

Que a vida non dea para entender todo non é unha frase feita: pódese estimar. Unha persoa, lendo durante toda a súa vida, non pasa duns dez xigabytes de texto. E é un tope xeneroso: supón ler moitas horas ao día durante décadas.

Para as contas deste libro abonda con pensar que cada letra ocupa un byte, así que dez xigabytes son arredor de dez mil millóns de caracteres. Un libro de trescentas páxinas, máis ou menos coma este que tes na man, anda polos seiscentos mil caracteres, así que estamos a falar dunhas **dezaseis mil obras**: uns trescentos metros de andel. Moitísimo máis do que unha persoa le de verdade na vida.

E hai outra maneira de dicilo que impresiona máis. Sumando o lido e o escrito, unha vida enteira de palabras cabe **nunha memoria USB das baratas**. Comprimida, sóbralle sitio.

Compárao co que veremos no capítulo 15. O GPT-3, de 2020, leu arredor de *1,2 billóns* de caracteres: unhas cento vinte vidas enteiras de lectura. Os modelos de 2024, unhas seis mil vidas. Nese desaxuste está boa parte do que explica este libro.

E o risco non está no texto: está no que facemos con el. Unha conta mal feita faiche pagar de máis no supermercado; a mesma conta mal feita nun informe bancario converte alguén con diñeiro en alguén sen el e un dato mal copiado nun informe médico pode converter unha persoa enferma nunha persoa morta. O que vai fóra da máquina son sempre decisións humanas.

Hai unha fronteira que convén ter marcada. Con moitos trebellos abonda con saber *usalos* e ninguén che vai pedir contas. Pero cando un trebello empeza a producir *afirmacións* («isto é certo», «esta candidatura é mellor», «este texto escribíchelo ti ou non»), saber usalo deixa de ser suficiente, porque para xulgar unha afirmación hai que saber de onde vén. E as intelixencias artificiais xenerativas son, exactamente, máquinas de producir afirmacións que non se sabe de onde veñen.

MÍRAO *cando si queremos saber?*

Hai moitas veces nas que si queremos saber. Cando no centro de saúde che dan un diagnóstico, moita xente pregunta «e como o saben?». Cando no taller che din que hai que cambiar a caixa de cambios, case todo o mundo pregunta «e por que?». Ninguén considera iso unha falta de respecto: é o xeito normal de decidir canto nos fiamos.

Diante dunha resposta xerada por unha máquina, esa pregunta case non se fai. En parte porque non se sabe a quen facerlla. En parte porque a resposta chega escrita con moitísima seguridade e a seguridade, en nós, adoita ser sinal de competencia. Neste libro veremos por que, coas máquinas, ese sinal non vale.

Os dous xeitos de non explicar

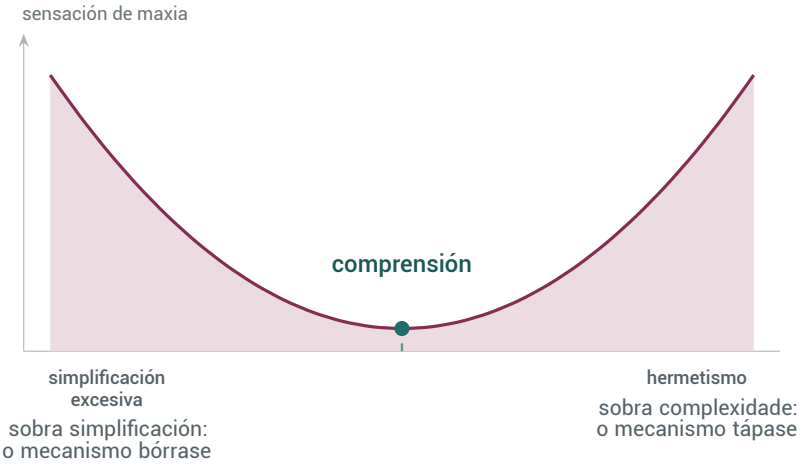
Se se quere que alguén non saiba como funciona algo, hai dous camiños fáciles. Son opostos entre si e, curiosamente, levan os dous ao mesmo sitio.

O primeiro é a **simplificación excesiva**. «A máquina pensa coma nós.» «Aprende soa.» «Le todo internet e despois contéstache.» Non son frases falsas de todo e ese é o problema: quedan tan curtas que o mecanismo desaparece. Quen as escoita non recibe unha explicación reducida, recibe unha imaxe e ademais unha imaxe enganosa. É o camiño que suxiren as propias palabras, porque dicimos que o programa *aprende*, que *entende*, que *alucina*, que *presta atención*. Todas esas palabras están tomadas emprestadas de nós. Ao longo do libro trataremos de evitalas, pero cando aparezan marcarémolas así, *en cor e en cursiva*, para que non se che esqueza que son **metáforas**.

O segundo é o **hermetismo**. «Baséase nun principio moi complicado.» «Ten moitas partes que se relacionan entre si.» «É un modelo xeneralista preadestrado con atención multicabeza.» Aquí a explicación existe, pero vai tan cargada que non hai por onde collela. A conversa péchase igual de ben ca co outro camiño, coa vantaxe de soar sobria e técnica. Quen o di tampouco adoita saber explicar como se pasa de contar palabras a escribir un soneto razoábel.

E o resultado dos dous é o mesmo: a **maxia**. Arthur C. Clarke deixouno escrito nunha frase que se cita moito: «Calquera tecnoloxía suficientemente avanzada é indistinguíbel da maxia» [29]. A frase é certa polos dous lados. Ante o hermetismo, ninguén entende porque a explicación non lle chega. Ante a simplificación exce-

siva, calquera cre que entendeu e o que ten na cabeza é un debuxo animado. Nos dous casos acábase atribuíndolle á máquina intencións, vontade, comprensión e, en versións máis ambiciosas, conciencia.



Convén facer un matiz, porque a imaxe pode enganar. O contrario do hermetismo non é simplificar: é a claridade. Unha explicación pode ser moi complexa e estar perfectamente clara, igual que pode ser moi sinxela e resultar completamente escura. O que teñen en común os dous extremos non é a cantidade de detalle, senón outra cousa: **nos dous pérdese a información que fai falta para comprender**. Nun agóchase; no outro, elimínase.

E o certo é que os sistemas de intelixencia artificial *si* son complexos: teñen moitas capas, moitos módulos e moitas partes que se relacionan entre si. Pero complexo non quere dicir incomprensíbel. A enxeñaría que os constrúe funciona precisamente por **abstracción**: dividir un sistema en capas, cada unha cunha función, para poder falar dunha sen ter que manter todas as demais na cabeza.

É como mirar un mapa a distintas alturas. Desde moi lonxe ves o conxunto e non os detalles; desde moi preto, os detalles e non o conxunto. Entre medias hai niveis intermedios e o que se busca é entender cada nivel no seu punto e como encaixa co de arriba e co de abaixo. Iso é simplificar, si, pero cunha simplificación ordenada, que é exactamente o contrario de agochar. Entender non é coñecer todos os detalles.

Os dous camiños cara á maxia cumpren a mesma función: **facer invisíbel o mecanismo**. E os dous teñen consecuencias prácticas. Quen quedou coa versión

simplificada fíase de máis: acepta unha resposta porque está ben escrita. Quen quedou diante do hermetismo renuncia antes de empezar: dá por feito que iso non lle atinxe e queda sen argumentos cando lle toca discutir en serio sobre qué se debe permitir e qué non. Hai tamén quen reacciona ao revés e o despacha con «non é máis ca un autocompletar», que soa crítico pero tampouco explica nada.

EN PALABRAS SINXELAS *un exemplo de explicación accesíbel*

Imaxina un espazo baleiro e ás escuras. Nun punto acéndese unha lámpada. A luz propágase en forma de onda e nós vémola en canto chega aos nosos ollos. Se non chega á retina non hai nada que ver: o espazo segue escuro para nós. E hai máis: se acendemos e apagamos a lámpada con certo ritmo, por esa mesma canle pódense mandar mensaxes en código Morse, con puntos e raias de luz.

Coas ondas de radio pasa algo parecido, coa diferenza de que os nosos ollos non son sensíbeis a esas frecuencias. Unha antena fai o papel de órgano sensor: cando unha onda chega ata ela, o seu campo eléctrico fai oscilar os electróns da antena e produce un pequeno sinal eléctrico que o receptor amplifica e interpreta.

Así, nun espazo que a nós nos parece baleiro e escuro pode estar pasando unha cantidade enorme de ondas: radio, televisión, telefonía móbil, Wi-Fi, Bluetooth, GPS, que non son máis ca outros códigos e outros protocolos facendo o mesmo que o Morse. Nós non as vemos; as antenas detéctanas.

E con isto acabouse a maxia e o misterio. Non quedaches sabendo deseñar unha antena, pero xa non precisas crer en nada: sabes que hai unha onda, algo que a emite e algo que a recolle. Ese nivel de explicación é o que este libro quere darche sobre as máquinas que escriben.

Hai unha terceira posición, a deste libro, que non é intermedia: é doutro tipo. Consiste en saber, con detalle suficiente, *qué operacións* se executan entre a túa pregunta e a resposta que aparece na pantalla. Cando sabes iso, xa non tes que decidir se crer ou non crer: tes criterio, que é outra cousa.

Por que agora si importa?

Para que saber como funciona a intelixencia artificial?

Podería obxectarse, con razón, que tampouco entendemos o motor do coche e non pasa nada. Pero fíxate na diferenza. O coche quérelo para desprazarte e, mentres te leve a onde queres de xeito seguro, como funcione dáche igual (iso si, sempre que non che dea rodeos que non veñen a conto). Nese sentido, un coche que se conduza só sería perfecto: abonda con que nos deixe no sitio.

A unha intelixencia artificial xenerativa pedímoslle outra cousa: que nos axude a preparar, a revisar, a comparar, a decidir. Aí xa non abonda con chegar: importa que o que nos dea sexa certo e completo. E como nos aseguramos de que o é? Esa pregunta non se responde mirando a pantalla.

A diferenza de fondo está nunha propiedade que ata hai moi pouco non tiña ningunha máquina: **podes manter con ela un diálogo**. Preguntas ou pides algo e devólveche unha resposta que a ti che parece razoábel; e outra vez tras outra. Por iso o que sae parécese ao pensamento.

E hai unha segunda propiedade que agrava a situación: **a maior parte das veces dálle a razón á persoa usuaria**. Se lle expós unha idea con convicción, o máis probábel é que a apoie e lle busque argumentos; se lle corrixes un dato que estaba ben, é bastante probábel que se desdiga e che pida desculpas. Un coche non che dá a razón. A interlocución con quen case nunca te contradí é agradábel e precisamente por iso é perigosa. No capítulo 10 veremos de onde vén ese costume e non é casualidade: **ese xeito de responder foi recompensado** mentres se axustaba o modelo, porque as persoas que puntuaban as respostas preferían as que lles daban a razón.

Un coche que funciona mal fai un ruído raro. Unha neveira que falla quéntase. Un texto xerado por unha máquina que se equivoca de medio a medio ten exactamente o mesmo aspecto que un que acerta: a mesma sintaxe impecábel, o mesmo ton seguro, as mesmas comas ben postas. Non hai ruído raro. Por primeira vez temos un aparello cuxos fallos *non se notan*: non hai ruído, nin cheiro, nin fume, nin unha luz que se acenda. O único indicio que usamos habitualmente para detectar un erro, que a cousa soe mal, desapareceu do deseño.

PÉNSAO UN MOMENTO *decidimos antes de pensar?*

Hai unha peza incómoda que convén poñer enriba da mesa, porque explica parte da nosa vulnerabilidade.

Boa parte das nosas decisións non se toman despois de razoar: tómanse antes, por vías rápidas e pouco conscientes; o razoamento chega despois para xustificalas. Hai experimentos clásicos nos que as persoas explican con toda seguridade por que escolleron unha cousa, dando razóns que demostrabelmente non foron as que influíron [107]. E hai modelos enteiros da psicoloxía moral construídos sobre esa orde: primeiro a intuición, despois os argumentos [62].

Non é un defecto de fábrica: é o que nos permite vivir sen pararnos a deliberar cada xesto. Pero ten unha consecuencia para o que nos ocupa. Se tendemos a decidir primeiro e argumentar despois, unha máquina que produce argumentos ben escritos para calquera cousa é exactamente a ferramenta que mellor encaixa nese oco. Dános o que xa iamos buscar.

Non o damos por demostrado; é unha hipótese razoábel que deixamos aquí para que a teñas presente ao ler o resto.

OLLO! *a confianza mal dirixida*

Cando lemos, xulgamos moi rápido e sen darnos conta: se un texto está ben construído, ten estrutura, usa o vocabulario preciso e non dubida, dámoslle crédito. Ese atallo funcionou durante séculos porque escribir así custaba traballo e o traballo era sinal de que alguén sabía do que falaba.

Ese vínculo rompeuse. Hoxe a forma impecábel é barata e a verdade segue sendo cara. O atallo quedou avariado e non hai xeito de desactivalo por dentro: só se pode compensar sabendo como se produciu o texto que tes diante. Non porque iso che diga se é certo, que non cho di, senón porque che di *onde* é máis probábel que falle e que che toca comprobar a ti.

A isto engádeselle unha cuestión de xustiza. Nun texto anterior, o primeiro autor e F. J. Fernández Fernández xa sinalaron que a desigualdade no acceso a estas ferramentas non se ve [9]. Cando a alguén lle faltan medios, adoita notarse: se non ten un bo ordenador ou acceso a bibliografía actualizada, vese no traballo e pódese ter en conta. Coa intelixencia artificial non pasa iso. As persoas que non

poden pagar unha subscrición entregan un documento coa mesma aparencia cós demais, só que algo peor; quen o avalía non ten como saber por que e conclúe, simplemente, que esas persoas son algo peores. Entender como funcionan non arranxa iso por si só, pero é a condición previa para poder discutilo: quen non sabe que diferenza hai entre unha versión de pago e unha gratuíta non pode nin reclamar nin lexislar sobre iso. As decisións sobre quen paga, quen accede e que se pode esixir tómanse igual, saibamos ou non; a diferenza é se estamos na conversa ou fóra dela.

E hai, por último, unha razón para querer saber que non ten que ver nin co medo nin co interese e que vale para este caso igual ca para moitos outros. Sinxelamente: **é moi interesante.** O que hai dentro desas máquinas é unha das construcións máis curiosas que fixo a nosa especie: unha maneira de converter o significado en xeometría, a sintaxe nunha rede e facer que miles de millóns de operacións moi simples produzan algo que se parece a unha conversa. Sería unha mágoa vivir rodeado disto sen mirar nunca dentro.

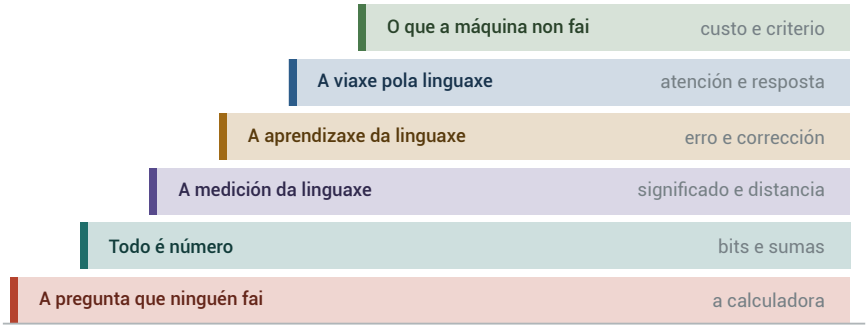
O camiño: empezamos pola calculadora

A tentación, ao escribir un libro coma este, é comezar polo final: empezar coa palabra *transformer*, debuxar unhas frechas e confiar en que se entenda. Non funciona. Segues as frechas e non entendes nada.

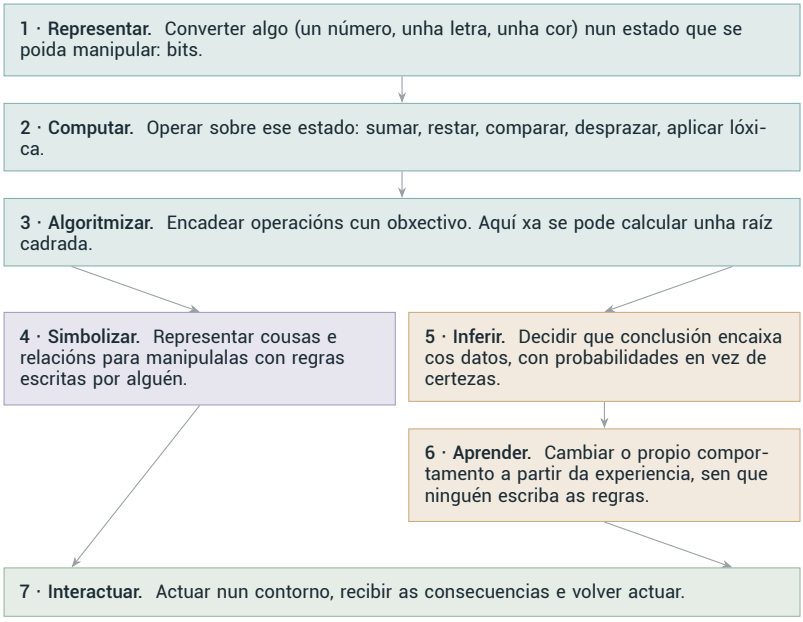
Aquí facemos o contrario. Empezamos por unha máquina que ninguén considera intelixente, a calculadora do teu peto, porque nela se poden ver, en pequeno e sen misterio, *as tres ideas que despois o explican todo*:

1. Unha máquina pode *dar* respostas correctas **sen saber nada**, porque efectivamente non sabe nada: só precisa un procedemento e alguén que o escribise.
2. Unha operación complexa pode resolverse cun procedemento que **repite operacións sinxelas moitas veces**.
3. O procedemento pode ser **aproximado** e para a máquina iso non supón nada: limítase a seguilo. Iso non quere dicir que non se poida saber canto se afasta: o erro pódese estimar e nos sistemas críticos é obrigado facelo.

Se cando remates o capítulo 1 tes clara a raíz cadrada de dous, xa entendiches o esqueleto de todo o demais. O resto do libro é subir chanzos sobre ese esqueleto.



Convén dicir tamén que son os chanzos. Non son só os bloques: son **niveis de abstracción**, capacidades que se van apoiando unhas nas outras. Esta é a escaleira enteira, a mesma que subiu a informática en cento cincuenta anos e a mesma que sobe este libro:



Dúas advertencias sobre este debuxo, porque unha escaleira tan limpa sempre mente un pouco.

A primeira: **cada chanzo non elimina o anterior**. Cando un modelo de linguaxe escribe unha frase segue a representar en bits, segue a computar sumas e segue

a executar un algoritmo. Non se substituíu nada; engadiuse enriba.

A segunda: **despois do terceiro chanzo a cousa bifúrcase**. Durante décadas houbo dúas tradicións distintas. Unha, a simbólica, escribía as regras a man: se tal cousa, entón tal outra. A outra, a estatística, prefería estimar e aprender dos datos. Non foron dous pasos consecutivos, senón dous camiños que competiron. O interesante é que volven xuntarse arriba: as máquinas destes anos aprenden como a segunda tradición e, ao usalas, parecen facer cousas da primeira.

Este libro sobe a escaleira enteira. Onde está cada chanzo:

chanzo	onde se ve	capítulos
1 · Representar	bloque I	3–4
2 · Computar	bloques 0 e I	1, 5
3 · Algoritmizar	bloque 0	1–2
4 · Simbolizar	bloque II	6–8
5 · Inferir	bloques III e IV	9, 12–13
6 · Aprender	bloque III	10–11
7 · Interactuar	bloque V	15–16

UN POUCO DE HISTORIA *cando se desenvolveu cada cousa?*

As datas aproximadas, para situarse:

cando	*que aparece*
desde sempre; moito máis no s. XIX	***representar e calcular**: ábaco, teares de Jacquard, tarxetas perforadas (censo dos Estados Unidos de 1890), calculadoras mecánicas*
1930–1950	***executar algoritmos**: Turing, Church, Zuse, von Neumann; primeiros ordenadores programábeis*
1950–1980	***manipular símbolos**: lóxica, linguaxes de programación, sistemas expertos*
desde sempre; con forza desde 1950	***inferir**: estatística, recoñecemento de patróns, decisión con incerteza*
1958–hoxe; disparado desde 2010	***aprender dos datos**: perceptrón, redes neuronais, moitos datos e tarxetas gráficas*

Pero coidado coa orde: **non é perfectamente cronolóxica**. A estatística é anterior aos ordenadores e as primeiras redes neuronais son anteriores ao auxe da intelixencia artificial simbólica. A escaleira ordena *capacidades*, non anos.

Aí está, tamén, o título deste libro. Nos primeiros capítulos a máquina só pode sumar: literalmente, non pode facer outra cousa. Nos últimos escribe, resume, traduce e argalla respostas que ninguén lle ditou. E entre uns e outros non se lle engadiu ningunha capacidade nova, ningunha faísca, ningún ingrediente secreto. Só **máis do mesmo, mellor organizado**, se ben a unhas escalas moitísimo maiores: como se pasásemos de facer castelos de area a construír o rañaceos máis alto que puidésemos imaxinar. Que iso abonde é a parte difícil de crer. Pero non escribimos este libro para que nos creas, senón para que o vexas ti.

Cada bloque responde unha pregunta que o anterior deixou aberta. Se dentro só hai números, como entran aí as palabras? Se as palabras son números, cales? Se ninguén lle dixo cales, de onde os sacou? E se os sacou ela soa, como consegue que a resposta veña a conto? Lendo seguido, a cousa vai crecendo soa.

⧖ PÉNSAO UN MOMENTO *antes de empezar*

Antes de pasar de páxina, proba a responder isto, aínda que sexa mal e apúntao nalgunha parte: *que cres ti que pasa entre o momento en que lle escribes unha pregunta a un destes programas de intelixencia artificial e o momento en que aparece a primeira palabra da resposta?*

Non hai trampa. Non importa se acertas. Importa que teñas unha resposta escrita, porque así, ao final do libro, vas poder comparar. É a mellor maneira de notar que entendiches algo.

As matemáticas que fan falta

Aquí cómpre honestidade, porque esta é a parte na que na maioría dos libros se mente un pouco.

Hai libros que prometen explicar isto «sen matemáticas». Pódese facer e déixanche a sensación agradábel de ter entendido. Pero o que entendes é unha historia *sobre* o mecanismo, non o mecanismo. Estas máquinas son obxectos matemáticos; explicalas sen matemáticas é como explicar unha sinfonía sen falar de son ou o parchís sen falar dos dados.

Entón, canta matemática? **A que se dá en cuarto da ESO.** Esa é a base sobre

a que está escrito todo o libro: sumar, restar, multiplicar e dividir; fraccións e porcentaxes; unha ecuación sinxela; coordenadas nun eixe; proporcións; algunha gráfica; e a idea de probabilidade que che deu para xogar á lotaría. Nada máis. Se deixaches as matemáticas ao rematar a ESO, porque escolliches un bacharelato sen elas ou porque te puxeches a traballar, ese é exactamente o punto de partida deste libro e non se supón nada que estea por riba.

Agora ben: **a partir de aí introducimos cousas novas**, sen présa e unha de cada vez. Non porque non se poida evitar, senón porque sería unha falta de respecto evitalo. Se estás lendo un libro enteiro sobre como funcionan estas máquinas, é porque tes interese e o interese é a mellor condición que existe para aprender algo. Sería absurdo non aproveitar ese interese.

Cando apareza unha peza nova, aparecerá así:

UN SÍMBOLO NOVO *como se presentan aquí os símbolos?*

Cada símbolo novo ten o seu cadro. Dise de onde vén, por que fai falta, que quere dicir exactamente e como se le *en voz alta*, que é a parte que case ningún libro conta e que máis axuda. Por exemplo: cando apareza x_1, x_2, x_3, non hai nada que memorizar; é a maneira de poñerlle unha orde numérica a unha lista de cousas do mesmo tipo e lese «xe sub un, xe sub dous, xe sub tres». Nada máis.

Non hai ningún símbolo neste libro que non se presente antes de usarse.

E cando queiramos subir un chanzo por riba do necesario (porque a cousa se pon fermosa e sería unha mágoa non contala), avisarase e poderás saltalo sen perder nada:

UN CHANZO MÁIS *que fas con estes cadros?*

Os cadros *Un chanzo máis* conteñen a versión un pouco máis fonda dunha idea que xa quedou explicada no texto. Son opcionais de verdade: o fío do libro non pasa por eles.

Se estás a gusto, entra. Se estás cansa ou canso, sáltao e segue; podes volver despois e o capítulo seguinte vaise entender igual. Non é trampa saltar: é lectura intelixente.

♥ **SEN MEDO** *se hai anos que non ves unha fórmula*

Imos dicir algo que convén ter presente en todo o libro: **a dificultade das matemáticas que aparecen aquí non está nas contas**. Ningunha conta deste libro é máis difícil que dividir. O que si custa é outra cousa: acostumarse a que un símbolo sexa un nome curto para unha idea longa.

Iso non se consegue lendo máis rápido. Conséguese lendo con lapis. Se algunha páxina se resiste, o problema case nunca é que sexa difícil: é que vai demasiado apertada para lerse coma unha novela. Copia o exemplo a man, fai as contas ti e adoita abrirse.

E se aínda así non se abre, salta e segue. Case todo se entende dúas veces: a primeira, a medias; a segunda, ben.

O que este libro non é

Por simetría, convén dicir tamén o que non vas atopar neste libro.

Non é un manual de uso. Non hai unha lista de instrucións máxicas para conseguir mellores respostas. Ese tipo de listas caducan en meses e ademais sobran: cando entendes o mecanismo, as boas instrucións dedúcense soas.

Non é unha profecía. Non imos dicirche se dentro de dez anos haberá máquinas conscientes nin cantos empregos desaparecerán. Non o sabemos e quen o afirma con seguridade, tampouco. O que si podemos facer é darche o suficiente para que distingas, cando a escoites, unha previsión argumentada dunha campaña comercial.

Non é un xuízo moral, aínda que teñamos opinións e as digamos cando toque. O bloque final trata sobre o que custa todo isto (enerxía, auga, datos alleos, traballo mal pagado) porque forma parte de como funciona, non como apéndice ético. Pero a conclusión sobre que facer con iso é túa.

E non é un libro completo. Hai partes (o adestramento real dun modelo grande, a arquitectura fina, as matemáticas de optimización) que aquí só se abordan ata onde se poden explicar sen caer en ningunha das dúas trampas: nin na da maxia nin na da frase curta. Cando cheguemos a un límite, dirémoscho.

O QUE SE DI POR AÍ *«ninguén sabe como funcionan, nin os que os fan»*

Esta frase óese moito e mestura dúas cousas moi distintas.

O mecanismo coñécese por enteiro, ata o último detalle: está escrito, publicado e pódese reproducir. Eses son os capítulos deste libro e é coñecemento público desde 2017 [160].

O que non se sabe é outra cousa: *por que un modelo concreto, con eses centos de milleiros de millóns de números concretos dentro, responde o que responde a unha pregunta concreta*. Iso non se sabe, pero non por razóns de segredo industrial: hai unha disciplina enteira dedicada a intentar descubrilo. É como coa neve: sabemos perfectamente como se forma un cristal de xeo e non podemos predicir a forma exacta da próxima folerpa.

Saber o primeiro sen saber o segundo non é saber a medias. É saber o que se pode saber (que é moitísimo) e é o que che vai dar criterio.

Unha nota sobre a lingua

Este libro está escrito en galego e non é un detalle decorativo. A meirande parte do que se publica sobre estes temas está en inglés e unha boa parte do resto en castelán. Escribir en galego sobre redes neuronais e sobre os circuítos de Shannon é unha forma modesta de normalización: dicir que esta lingua serve para isto, tamén, porque serve para todo.

E tampouco é unha novidade que desde aquí se pense en máquinas de calcular. Hai un nome que case non se cita e que ademais vén a conto do capítulo co que abrimos.

UN POUCO DE HISTORIA *un home de Curantes e a primeira máquina que multiplicaba de verdade*

Ramón Silvestre Verea García naceu en 1833 en Curantes (A Estrada) e morreu en Bos Aires en 1899. Polo medio foi emigrante, xornalista e editor en Nova York e o 10 *de setembro de* 1878 *recibiu a patente estadounidense número* 207 918 *por unha máquina de calcular [161].*

Non era unha máquina de calcular máis. Case todas as anteriores multiplicaban do único xeito que sabían, sumando unha e outra vez: para facer 7 × 8 *había que darlle oito voltas á manivela. A de Verea multiplicaba directamente e o truco é tan bonito que paga a pena contalo: dentro levaba dous prismas de latón de dez caras, furados con buratos de diámetros distintos. Eses buratos* ***eran*** *a táboa de multiplicar, feita de metal. A máquina non calculaba* 7 × 8*: ía buscar onde estaba gardado.*

Garda iso, porque é o capítulo 1 enteiro por adiantado. Unha máquina pode dar respostas correctas sen saber nada, se alguén lle deixou preparado o procedemento (ou, neste caso, a resposta) por diante.

O máis galego da historia é o final. Contan as crónicas que rexeitou as ofertas para comercializala e que dixo que non a fixera polo diñeiro senón para demostrar que alguén de aquí podía inventar igual ca un americano.

E non é unha lenda que se conte de boca en boca: o modelo está nun museo e ten número. Consérvase no National Museum of American History do Smithsonian, en Washington, catalogado como Verea Calculating Machine Model co número 311 942 *[161]. Mide* 13 *por* 17 *por* 20 *centímetros, ten os lados de latón e está* ***aberto por diante, por detrás e por abaixo****, a propósito, para que se lle vexa o mecanismo. Que é, exactamente, o que imos facer nós durante todo o libro.*

Iso obriga a decidir vocabulario e a regra que seguimos para iso ten tres chanzos. Primeiro: se existe unha palabra galega clara, úsase. Segundo: se non existe, buscámola en portugués, que ten terminoloxía técnica moderna asentada; parécenos preferíbel iso a calcar do castelán ou a deixar o anglicismo por defecto. E só cando non hai nin unha nin outra se mantén o termo orixinal (que adoita ser un nome propio internacional) e explícase. Por iso aquí lerás *rede neuronal*, *atención*, *peso* ou *incrustación* e tamén lerás *transformer* ou *softmax*, porque non ten sentido inventar unha palabra nova para algo que todo o mundo vai atopar escrito así. Ao final do libro hai un glosario con todas.

Sobre isto hai unha obxección que aparece sempre e que convén desactivar antes de empezar: a de que tal palabra «non existe en galego». Hai anos, nun debate sobre patrimonio cultural inmaterial, o primeiro autor preguntoulle a un lingüista se un determinado termo existía ou non. A resposta foi: non te preocupes, tampouco existen os ovnis e ben que escribimos deles.

Lévao posto durante todo o libro. As palabras non existen antes de usarse: existen porque alguén as usa e despois xa non hai quen as quite.

E hai unha razón máis para escribir isto en galego, que non é sentimental e que se entende enteira no capítulo 8. As intelixencias artificiais xenerativas aprenden do que está escrito. En formato dixital e libre hai pouco texto en galego, moito menos do que nos gustaría. Iso fai que estas máquinas respondan peor na nosa lingua e que procesala lles resulte máis custoso. Non hai ningunha solución técnica para iso: a única maneira de que haxa máis texto en galego é que alguén o escriba.

Así que este libro é tamén, moi modestamente, unha liña máis nesa conta. **A ciencia é importante. A ciencia en galego tamén.**

UN POUCO DE HISTORIA *a palabra «computador» designou primeiro persoas*

Durante case dous séculos, un computer era unha persoa. Nos observatorios astronómicos, nas oficinas de cálculo balístico e nos equipos que preparaban táboas de navegación traballaban centos de persoas, moitas delas mulleres, que executaban á man, unha tras outra, as operacións dun procedemento que outra persoa deseñara. Non tiñan que entender a finalidade do cálculo: tiñan que seguir as regras sen erros.

Convén non esquecelo, porque é a idea central da informática antes de que houbese máquinas: se un procedemento está ben especificado, executao calquera, ou calquera cousa, sen necesidade de comprendelo. Todo o demais son detalles de velocidade.

Unha última cousa

Volvemos á aula do principio.

O que estaba en xogo naquela pregunta non era a calculadora. Era comprobar se seguía viva a capacidade de estrañármonos diante dun obxecto cotián: esa forma de mirar algo que usaches mil veces e pensar «espera, isto como pode ser?». Calquera cativo ou cativa de tres ou catro anos pasa o día preguntando *por que*: unhas veces buscando a causa e outras a finalidade. Ese motor, curiosamente,

apágase arredor da escolarización. O primeiro autor argumentou noutro lugar que a escola, tal e como está montada, tende a erosionar precisamente esa capacidade, porque a pregunta espontánea se converte nunha interrupción [4]. Seguimos pensando que é certo e pensamos tamén que a capacidade non se perde: adormece.

Se algo queremos que faga este libro, é iso: espertala un pouco. Que, ao rematar, non só saibas como funcionan estas máquinas, que tamén, senón que che volva a dar por preguntar como funcionan as outras. O ascensor, a neveira, o buscador ou o semáforo que sabe que estás aí.

Ninguén pode entender todo. Pero todo o mundo pode entender algo e a sensación de entender algo por dentro, unha soa vez, cambia a relación co resto. Empecemos pola raíz cadrada de dous.

Como ler este libro

O libro está pensado para lerse seguido, de principio a fin, porque cada bloque contesta unha pregunta que deixou aberta o anterior.

Antes, un mapa rápido. O bloque 0 funciona como un prólogo: abre unha calculadora e amosa, nun caso pequeno, como traballan estas máquinas por dentro. O bloque I conta de que están feitas: números en binario e portas lóxicas. Con iso xa se pode empezar a tratar a linguaxe no bloque II e, a partir de aí, cada bloque sobe un chanzo.

Dito isto, hai varias maneiras razoábeis de percorrelo.

Enteiro. Do capítulo 1 ao 16, sen saltar. É como está escrito e é o que recomendamos: cada peza usa as anteriores.

Sen o prólogo. O bloque 0 pódese saltar: empeza polo capítulo 3, no bloque I. Cando queiras ver de onde saen as tres ideas que o explican todo, volve á calculadora.

Con présa. Capítulos 1, 3, 7, 9, 12, 13 e 14. Perdes exemplos e historia, pero o fío do mecanismo queda completo.

Polo final. Se o que che urxe é decidir como usar isto, le o bloque V primeiro e volve despois ao principio.

O único que non recomendamos é empezar polo capítulo 12, que é onde está a peza famosa. Sen os anteriores, lese e non se entende.

Os cadros de cor

Ao longo do libro hai cadros de cores. Cada cor quere dicir unha cousa e saber cal che permite decidir sobre a marcha se paras ou segues.

Cadro	Que hai dentro
En palabras sinxelas	A idea sen tecnicismos, cunha comparación ou unha imaxe mental. Se só les os cadros azuis, sacas o argumento completo do libro, aínda que sen o mecanismo.
Por que funciona?	A xustificación de verdade: por que ese procedemento dá o resultado correcto e non outro. É a parte que distingue entender de crer.
Exemplo	Un caso concreto resolto paso a paso, con números pequenos, para que poidas seguilo a man.
Mírao	A mesma idea, pero vista: un debuxo, un esquema, unha maneira de imaxinala.
Ollo!	Un erro de razoamento que comete case todo o mundo neste punto e por que se comete. Se levas anos usando estas ferramentas, presta atención a estes.
O que se di por aí	Unha afirmación moi estendida sobre a intelixencia artificial, posta fronte ao que sabemos. Ás veces é falsa; máis veces é medio certa, que é peor.
Un pouco de historia	De onde veu esta idea e a quen se lle ocorreu. Case nada do que hai nestas máquinas é de onte. Van en cursiva, para distinguilos do fío principal.
Un símbolo novo	A presentación coidadosa de cada peza de notación: que significa, por que fai falta e como se le en voz alta.
Un chanzo máis	Un pouco por riba do nivel do libro, para quen queira. **Opcional de verdade**: o fío non pasa por aquí.
Próbao ti	Algo para facer con lapis e papel (ou, ás veces, cun destes programas diante). Están numerados e teñen resposta ao final do libro.
Pénsao un momento	Unha pregunta sen resposta escrita, para deterse trinta segundos antes de seguir. Pararse é parte de ler.
Sen medo	Para cando a cousa aperta. Non son ánimos baleiros: son indicacións concretas sobre como saír do atasco.
Para que?	Para que serve saber isto, dito sen adornos. Se a resposta honesta é «porque é fermoso», tamén se di.

Hai ademais unhas marxes sen cor, cunha liña vertical ao lado esquerdo, que non son cadros para saltar senón o contrario: aparecen tres veces en todo o libro e cada unha recolle, en dúas liñas, algo que xa quedou explicado enteiro nas páxinas anteriores. Son as *Definicións* (algoritmo, descenso por gradiente, ou exclusivo). Se despois necesitas volver por algunha delas, están no índice e no glosario.

E ao final de cada capítulo hai un cadro da cor do bloque, *O esencial deste capítulo*, cunha lista do que convén levar. Se algún día volves ao libro só para buscar unha cousa, empeza por aí.

Tres consellos prácticos

Le con lapis. Non é unha frase feita. Os exemplos deste libro están escollidos con números pequenos precisamente para que os poidas facer ti nunha esquina do papel. A diferenza entre ler «a máquina multiplica duplicando» e duplicar ti cinco veces nunha columna é toda a diferenza que hai.

Ten unha destas máquinas diante. Varios cadros pídenche que lle fagas unha pregunta concreta a un programa de intelixencia artificial xenerativa e que mires ben a resposta. Vale calquera; non importa cal. O interesante non é o que responda, senón que despois de ler o capítulo correspondente vas saber *por que* respondeu iso.

Non teñas présa coas palabras prestadas. Cando vexas unha palabra marcada *así*, é un aviso: esa palabra tomámola emprestada do vocabulario humano e non significa aquí exactamente o que significa en nós. *Aprender*, *atención*, *alucinar*, *entender*. Son metáforas útiles e perigosas ao mesmo tempo, coma case todas as boas metáforas.

♥ **SEN MEDO** *canto se tarda?*

Cada capítulo ocupa entre quince e trinta páxinas, pensadas para unha ou dúas sentadas tranquilas. Non é un libro para ler nun día e tampouco se gaña nada facéndoo.

Se un capítulo che leva dúas semanas, vai perfecto. Se saltas un cadro *Un chanzo máis*, vai perfecto tamén. O único xeito de ler isto mal é lelo enteiro sen deterse nunca, chegar ao final e non ser quen de explicarlle nada disto a outra persoa.

BLOQUE 0

A pregunta que ninguén fai

Antes de abrir unha máquina que escribe, abrimos unha que suma. As dúas fan o que fan, pero ningunha o fai como parece que o fai.



CAPÍTULOS 1 E 2

Capítulo 01

Como o fai a túa calculadora?

«A máquina analítica non ten ningunha pretensión de orixinar nada. Pode facer calquera cousa que saibamos ordenarlle que faga.»

Ada Lovelace, 1843

PARA ARRANCAR

A calculadora do teu teléfono dáche a raíz cadrada de dous nun pestanexo, con quince decimais. Dentro non hai ningunha táboa de raíces. Tampouco hai ningunha táboa de senos, nin sequera a de multiplicar. Que hai entón e como consegue o resultado correcto?

1.1 Unha pregunta que parece parva

Colle unha calculadora (vale a do teléfono) e escribe 7×8. Responde 56. Ata aquí, nada.

Agora fai unha pregunta rara: *como o fixo?*

A resposta que dá case todo o mundo é que o ten gardado. E é unha resposta sensata, porque é o que fixemos nós: aprendemos a táboa de multiplicar de memoria, a forza de cantala. Cando nos preguntan canto son sete por oito non

calculamos nada: recuperamos. Parece razoábel supoñer que a máquina fai o mesmo, só cunha táboa máis grande gardada na memoria.

Se cadra xa che veu á cabeza unha obxección: aínda que esa táboa xigante existise, *como podería atopar nela tan axiña a entrada que busca*? Deixa a pregunta apuntada, que se responde dúas páxinas máis adiante.

Ademais, houbo unha máquina que o fixo exactamente así. A de Ramón Verea, a do proemio, levaba a táboa de multiplicar dentro, furada en dous prismas de latón: para 7 × 8 non calculaba, ía buscar o burato correspondente. Foi un invento brillante en 1878. O problema aparece en canto os números medran.

Proba entón con 3 947 × 8 821. A calculadora responde 34 816 487 e tarda o mesmo ca antes. Se tivese as respostas gardadas, tería que ter gardadas *todas*: só coas multiplicacións de dous números de catro cifras xa serían uns oitenta e un millóns de parellas e iso limitándonos a catro cifras.

O QUE SE DI POR AÍ *«a calculadora ten as respostas gardadas»*

É a idea máis estendida e é falsa, pero ten unha parte verdadeira que convén separar ben, porque a mesma confusión vai reaparecer ao falar de intelixencia artificial.

Falso: que exista dentro unha lista de resultados que se consulta. Non a hai e non podería habela: as combinacións posíbeis son infinitas e a memoria non o é. (No cadro seguinte hai números concretos.)

Certo: que dentro haxa *algúns* números gardados. Para calcular senos, por exemplo, a máquina si consulta unha pequena lista (está no apéndice A). Pero ten unhas vinte entradas, non oitenta e un millóns. A diferenza entre gardar vinte números e gardar todas as respostas non é de tamaño: é de natureza. Vinte números non son unha lista de respostas. Son *ferramentas*.

MÍRAO *canto ocuparía gardar a táboa?*

Fagamos a conta, porque é máis contundente ca calquera argumento.

Os números de catro cifras van do 1 000 ao 9 999: son 9 000. As parellas posíbeis son 9 000 × 9 000 = 81 000 000.

Oitenta e un millóns de resultados. Antes de pasar a bytes, imaxínao en papel: se os escribises en táboas apañadas, a oito columnas e sesenta filas por páxina, cada páxina levaría 480 números e farían falta

$$81\,000\,000/480 \approx 169\,000 \text{ páxinas},$$

é dicir, **arredor de catrocentos libros coma o que tes na man**. Só para as multiplicacións de catro cifras.

Agora en bytes. Un **byte** é a unidade na que se mide a memoria: son oito bits (capítulo 3) e pode gardar $2^8 = 256$ valores distintos. Con catro bytes as combinacións multiplícanse: $256 \times 256 \times 256 \times 256$, máis de catro mil millóns de valores distintos. O produto máis grande, $9\,999 \times 9\,999 = 99\,980\,001$, non chega aos cen millóns, así que cabe sobrado en catro bytes. Así que gardar só esa táboa ocuparía

$$81\,000\,000 \times 4 = 324\,000\,000 \text{ bytes}$$

Compárao con dúas máquinas reais:

máquina	bytes
HP-35, a primeira calculadora científica de peto (1972): o seu programa enteiro	960
Ordenador de guiado do Apolo (1969): programa e datos	77 824
Só a táboa de multiplicar de catro cifras	324 000 000

A táboa ocuparía **máis de catro mil veces** toda a memoria do ordenador que levou persoas á Lúa e máis de trescentas mil veces o programa enteiro dunha calculadora que xa facía senos e logaritmos. E iso só con catro cifras: cunha calculadora de dez, as parellas serían 81 trillóns. E non axuda moito aproveitar que 7×8 dá o mesmo ca 8×7: gardando só a metade da táboa seguirían facendo falta máis de corenta millóns de resultados.

A HP-35 cabía nun peto e podía facer moitísimo máis que multiplicar, cun programa de 960 bytes. Non tiña as respostas. Tiña receitas.

E aquí vai a resposta á pregunta que quedou apuntada. Unha táboa así serviría, pero quen non sabe multiplicar tería que percorrela de arriba abaixo ata dar co resultado: iso eternízase. (Se a táboa está ordenada hai xeitos de buscar moito máis rápido, partindo pola metade unha e outra vez, pero esa é outra historia.) E, sobre todo, o problema non é atopar. É **ter**. E unha táboa que non existe non hai onde buscala.

Se as multiplicacións non están gardadas, hai que producilas. E producilas quere

dicir: executar, unha tras outra, un número finito de operacións moi sinxelas. A palabra para iso é **algoritmo** e todo este capítulo consiste en mirar tres deles.

EN PALABRAS SINXELAS *o que é unha calculadora*

Unha calculadora é unha máquina que fai directamente, cos seus circuítos, unhas poucas operacións moi simples e que as fai a unha velocidade que non nos cabe na cabeza: miles de veces por segundo nunha de peto e centos de millóns na do teléfono.

Nas máis sinxelas esa lista é curtísima: **sumar, restar, comparar dous números e desprazar as súas cifras á esquerda ou á dereita**. Todo o demais (a raíz cadrada, o seno, o logaritmo, as potencias) non son capacidades adicionais: son receitas escritas con esas poucas operacións. Non hai dentro ningunha operación de raíz cadrada; o que hai é unha receita que, seguida ao pé da letra, acaba producindo unha raíz cadrada, ou, como veremos, unha aproximación dela.

PARA QUE? *empezar pola calculadora*

A calculadora é o caso máis limpo de algo que atoparemos cinco veces máis ao longo do libro, cada vez máis difícil de aceptar: **unha máquina pode dar respostas correctas sen saber nada**.

Ninguén se escandaliza cando o dicimos dunha calculadora. Cando o digamos dun programa que escribe cartas de recomendación, custará máis. E vai ser exactamente a mesma afirmación, coa mesma estrutura, só que con máis pasos. Convén telo visto antes nun caso onde non nos xogamos nada emocionalmente.

1.2 Multiplicar sen saber a táboa

Empecemos polo máis fácil. Supón que unha máquina só pode sumar, duplicar (que é sumar algo consigo mesmo) e partir pola metade quedando coa parte enteira. Con iso abonda para multiplicar calquera par de números.

A maneira máis rudimentaria de multiplicar 7×8 é sumar oito sete veces, ou sete oito veces. Funciona e é lentísimo: para $3\,947 \times 8\,821$ farían falta case catro mil sumas. Hai un xeito moitísimo mellor e ten máis de tres mil anos: xa está

no papiro Rhind exipcio. No exemplo 1.1 amósase como se multiplican dous números con metades e dobres, tal e como está escrito no papiro.

UN POUCO DE HISTORIA *sete por tres, na cociña dunha casa*

O terceiro autor conta unha escena que lle pasou hai anos na súa casa. Unha nena de seis anos acababa de aprender a sumar números pequenos. Unha de oito estaba na fase de aprender a táboa de multiplicar, cantándoa.

Preguntoulles ás dúas canto era 7 × 3. A maior precisou o seu tempo: tiña que atopar a táboa correcta na cabeza e despois a fila. A pequena respondeu 21 case sen pausa.

A maior, entre asombrada e amolada, preguntoulle como o fixera tan rápido. E a pequena contestou, sen darlle importancia: é moi fácil, é 7 + 7 + 7.

Esa é exactamente a discusión deste capítulo, en dúas frases e sen ningún ordenador diante. Unha buscaba nunha táboa e a outra executaba un procedemento. A táboa é máis rápida cando o caso está nela. O procedemento funciona sempre.

EXEMPLO 1.1 *multiplicar 23 × 45 duplicando*

Escribimos dúas columnas. Na esquerda poñemos 23 e imos partindo pola metade descartando o resto: 23 → 11 → 5 → 2 → 1. Na dereita poñemos 45 e imos duplicando. (A primeira e a última columna da táboa, en gris, só están para axudar a seguir o exemplo: na práctica abonda con riscar as filas nas que as metades son pares e sumar os dobres que quedan.)

fila	metades	dobres	súmase?
0	23	45	✓
1	11	90	✓
2	5	180	✓
3	~~2~~	~~360~~	✗
4	1	720	✓

A regra: márcanse as filas nas que o número da columna das metades é impar. Súmanse os números da columna dos dobres desas filas:

$$45 + 90 + 180 + 720 = 1\,035$$

E, en efecto, 23 × 45 = 1 035.

Catro metades, catro duplicacións e tres sumas, en vez de vinte e dúas sumas. E nin unha soa vez fixo falta saber ningunha táboa de multiplicar.

UN POUCO DE HISTORIA *o papiro Rhind*

O método de duplicar e sumar aparece no papiro Rhind, un rolo exipcio copiado polo escriba Ahmes arredor do 1550 a.C. a partir dun orixinal máis antigo, que é unha das principais fontes que conservamos sobre matemática exipcia.

Os exipcios non tiñan táboas de multiplicar: multiplicaban duplicando, exactamente así. Durante séculos chamóuselle en Europa «multiplicación rusa» ou «do campesiño ruso», por unha atribución tardía e bastante arbitraria. O que importa aquí é outra cousa: o procedemento que usa a túa calculadora para multiplicar dous números é, no fondo, o mesmo que usaba un escriba hai trinta e cinco séculos. Cambiou a velocidade. Non cambiou a idea.

Noutros sitios si había táboas. A táboa de multiplicar decimal máis antiga que se coñece é chinesa: está escrita en vinte e unha tiras de bambú de arredor do ano 305 antes da nosa era, que hoxe garda a Universidade Tsinghua. Organizada como unha cuadrícula, permitía multiplicar números enteiros e medios ata 99,5 [91, 120]. Se a cantaban ao memorizala, coma nós, non o sabemos.

POR QUE FUNCIONA? *por que a regra dos impares funciona?*

Non é un truco: é unha maneira de escribir o 23.

Como na táboa do exemplo, as filas van numeradas desde cero. Así, á fila k correspóndelle a potencia 2^k:

$$2^0 = 1, \quad 2^1 = 2, \quad 2^2 = 4, \quad 2^3 = 8, \quad 2^4 = 16$$

fila	potencia	metades	dobres	súmase?
0	$2^0 = 1$	23/1 → 23	45 × 1 = 45	✓
1	$2^1 = 2$	23/2 → 11	45 × 2 = 90	✓
2	$2^2 = 4$	23/4 → 5	45 × 4 = 180	✓
3	$2^3 = 8$	~~23/8~~ → ~~2~~	~~45 × 8~~ = ~~360~~	✗
4	$2^4 = 16$	23/16 → 1	45 × 16 = 720	✓

(A frecha quere dicir que se queda só coa parte enteira: 23/2 = 11,5 → 11. As columnas en gris son de axuda: na práctica só se escriben as metades e os dobres, ríscanse as filas pares e súmase o que queda.)

As filas marcadas son a 0, a 1, a 2 e a 4, é dicir, as que lles corresponden 1, 2, 4 e 16. A fila 3 queda riscada porque alí o número da columna das metades é par. E resulta que

$$1 + 2 + 4 + 16 = 23$$

Acabamos de **descompoñer o 23 nunha suma de potencias de dous** sen propoñérnolo: iso (e non outra cousa) é o que fai a columna das metades.

Mira agora a columna da dereita coa táboa diante. Non son números que saian de ningures: son **o 45 multiplicado por esas mesmas potencias de dous**, en orde crecente. E como duplicar é a operación máis fácil que hai, obtéñense unha detrás doutra sen multiplicar nada.

Recapitulemos as dúas columnas:

- **Na esquerda, as metades:** divídese entre dous e consérvase a parte enteira. Serve para descompoñer o 23.
- **Na dereita, os dobres:** multiplícase por dous. Serve para ter listos os produtos 45 × 1, 45 × 2, 45 × 4...

Unha vez feito iso, a multiplicación é inmediata: só hai que **sumar os valores da columna dos dobres nas filas en que a columna das metades é impar**, que son precisamente as potencias que fan falta:

$$23 \times 45 = (1 + 2 + 4 + 16) \times 45 = 45 + 90 + 180 + 720 = 1\,035$$

Dito doutro xeito: **a columna esquerda descompón, a columna dereita duplica e a suma final recompón**.

O paso que permitiu partir a conta en catro anacos ten nome: **propiedade distributiva**. Di que multiplicar unha suma é o mesmo que multiplicar cada sumando por separado e sumar despois:

$$a \times (b + c) = a \times b + a \times c$$

Con números: 3×12 pódese facer como $3 \times (10 + 2) = 30 + 6 = 36$. Ninguén o chama así, pero non é mala forma de multiplicar de cabeza.

O único fino é que a descomposición en potencias de dous sempre existe e é única, cousa que veremos no capítulo 3.

10 E hai un detalle que converte todo isto en algo máis ca unha curiosidade histórica. Dentro da máquina os números escríbense en base dous (capítulo 3) e alí **duplicar e partir pola metade son as operacións máis baratas que existen**: multiplicar por dous é desprazar todas as cifras un lugar á esquerda e dividir por dous é desprazalas un lugar á dereita. Exactamente igual que en decimal multiplicar por dez é engadir un cero. As dúas columnas do exemplo, as metades e os dobres, son dous desprazamentos.

A frase sobre a descomposición única merece que nos deteñamos, porque acabamos de tropezar coa primeira idea grande do libro sen darnos conta. **Todo número se pode escribir como suma de potencias de dous, dunha única maneira.** O 23 é 16 + 4 + 2 + 1 e non hai outra forma de facelo con potencias de dous distintas. Esa propiedade é a razón profunda de que os ordenadores usen ceros e uns e é o tema do capítulo 3.

PRÓBAO TI 1.1 ★☆☆

Multiplica 19 × 37 polo método das metades e os dobres. Escribe as dúas columnas, marca as filas impares e suma.

Despois comproba dúas cousas: (a) que o resultado coincide co que dá a calculadora; (b) que as filas que marcaches corresponden exactamente á descomposición de 19 en suma de potencias de dous.

resposta na páx. 365 →

UN CHANZO MÁIS *canto custa multiplicar de verdade?*

Partir, duplicar e sumar están moi ben, pero os ordenadores de hoxe non multiplican así e paga a pena saber por que. Depende de como se mida o custo dun procedemento, que é unha idea que volverá nos capítulos 12 e 15.

Está contado na sección A.2 do apéndice A, xunto cunha historia que vai de Babilonia a 2021 e que ten unha moral sorprendente: *quen ten memoria aforra cálculo*.

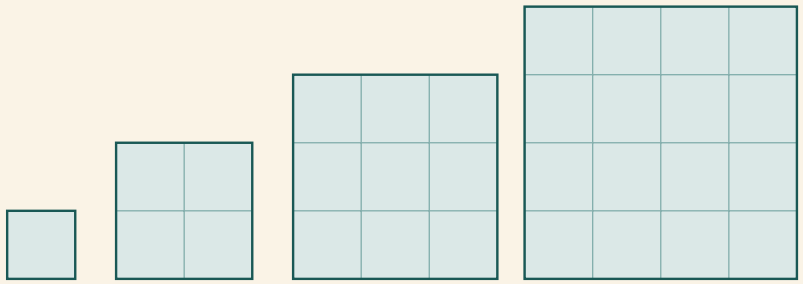
1.3 A raíz cadrada: estimar e corrixir

Coa multiplicación vimos unha receita enteira. Imos ver a segunda, que é a que máis nos vai servir despois, porque a idea que hai dentro é literalmente a mesma que usará a máquina do capítulo 10 para aprender.

Antes de calcular nada, convén ter diante o que é unha raíz cadrada. Non é unha operación misteriosa: é unha pregunta sobre cadrados.

MÍRAO *unha raíz cadrada é o lado dun cadrado*

Constrúe cadrados con cadradiños iguais, coma quen xoga con regretas:



Un cadrado de lado 4 está feito con $4 \times 4 = 16$ cadradiños. Iso é todo o que significa $4^2 = 16$. E a pregunta ao revés, *que lado ten un cadrado feito con 16 cadradiños?*, é exactamente $\sqrt{16} = 4$.

A raíz cadrada dun número é o lado do cadrado que ten ese número como área.

Agora o interesante. Cantos cadradiños fan falta para un cadrado de área 2? Con un quedas curto; con catro pásaste. Non hai cadrado de cadradiños enteiros con área 2. Pero existe e pódese debuxar:



O cadrado grande ten 4 cadradiños. Unindo os puntos medios dos lados queda dentro un cadrado inclinado: cada cadradiño queda cortado pola metade e o inclinado leva catro medias partes, é dicir, **área 2**.

O seu lado é a diagonal dun cadradiño. **Esa diagonal mide $\sqrt{2}$.**

$\sqrt{2}$ non é unha abstracción: é o que mide a diagonal dun cadrado de lado 1. Está en cada baldosa do chan.

Se queres $\sqrt{2}$, entón, estás a buscar o lado dun cadrado de área 2. Podes probar: 1,5 dá 2,25 e pasácheste; 1,4 dá 1,96 e quedaches por baixo. Podes seguir así toda a tarde e vas avanzando, pero moi amodo.

O método bo consiste en non probar ás cegas, senón **usar o erro para corrixirte**. E vese mellor debuxado.

MÍRAO *o rectángulo que quere ser cadrado*

Pensa en $\sqrt{2}$ como o lado dun cadrado de área 2.

Como cadrados exactos non os imos atopar tenteando, traballamos con **rectángulos de área 2** e imos facéndoos cada vez máis cadrados.

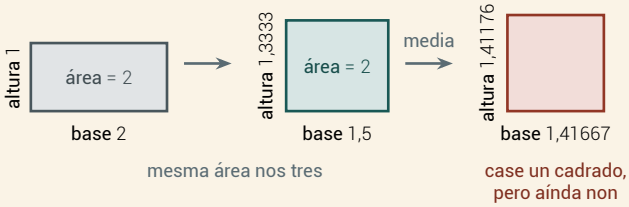
Empeza polo máis alongado de todos: base 2 e altura 1. Área 2, si, pero non se parece nada a un cadrado. Proba agora con base 1,5: como a área ten que seguir sendo 2, a altura é obrigatoriamente 2/1,5 = 1,3333**...**

Fíxate no que pasou: se un lado se pasa, o outro *queda curto*. Sempre. Non poden fallar os dous no mesmo sentido, porque o produto ten que seguir sendo 2.

E aquí está o truco: se un lado é grande de máis e o outro pequeno de máis, a súa *media* está máis preto do que buscamos có lado grande. Fagamos a conta enteira, sen saltar nada:

$$\frac{1{,}5 + 1{,}3333}{2} = 1{,}41667, \qquad \frac{2}{1{,}41667} = 1{,}41176$$

Con esa media facemos un rectángulo novo, de base 1,41667 e altura 1,41176: mesma área e moito máis cadrado có anterior. E repetimos.



Iso é todo o método e ten case catro mil anos: os babilonios xa o usaban e nos libros adoita chamarse *método de Herón*. **Colle un lado, mira canto mide o outro e queda coa media dos dous.** Repite.

paso	estimación	cifras correctas
0	1,5	1
1	1,416666…	3
2	1,4142156…	6
3	1,41421356237…	12

Cada volta **duplica, máis ou menos, o número de cifras correctas**. Por iso en moi poucos pasos xa hai máis precisión da que colle no visor e por iso este método, ou variantes moi próximas del, é o que levan dentro moitas calculadoras e non o outro.

Tres cousas que paga a pena levar. A primeira: a receita parte dunha estimación mala, pero iso non importa, porque permite *medir o erro e corrixila*. A segunda: iso é, palabra por palabra, o que fará a máquina do capítulo 10, só que corrixindo cen mil millóns de números en vez dun só. E a terceira: nunca remata. Párase.

UN CHANZO MÁIS *a raíz cadrada, enteira*

Este método ten unha historia longa, ao igual ca outro método que lle fai parella. No apéndice A están os dous.

Na sección A.3 está o outro método, o que se aprendía na escola: *separa, baixa, dobra, proba, resta*. Resulta que ten unha xustificación dunha liña que se pode **debuxar**, cunha figura que se chama *gnomon* e que explica de onde sae ese «dobra o que levas». Se algunha vez sufriches este cálculo de raíz cadrada sen que che dixesen por que, esa páxina é para ti.

E na sección A.4 están todas as contas do método que se ve neste capítulo, empezando por un valor calquera para ver que sempre converxe á raíz buscada, a demostración de por que a media non pode empeorar e a razón de que as cifras correctas se dupliquen. Alí tamén se ve que este método babilónico é, sen cambiarlle unha coma, un caso particular do procedemento xeral que Newton publicou no século XVII.

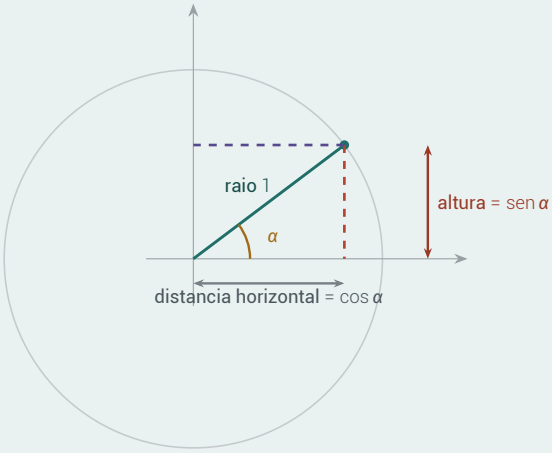
1.4 O seno: xirar no canto de medir

A raíz cadrada aínda se podía facer con multiplicacións e divisións. Pero cando lle pides á calculadora o seno de 37°, a cousa parece imposíbel: o seno defínese cun triángulo e dentro da máquina non hai triángulos.

UN SÍMBOLO NOVO *recordatorio: que era o seno*

Se hai tempo que non traballas co seno, abonda con ver esta imaxe; non fai falta máis.

Debuxa unha circunferencia de raio 1 e un raio que forma un ángulo α coa horizontal. O extremo dese raio é un punto. Pois ben: a **altura** dese punto é o *seno* de α e a súa **distancia horizontal** é o *coseno* de α.



Así, calcular o seno de 37° é o mesmo que preguntar: *se saio do punto* (1,0) *e xiro* 37 *graos, a que altura quedo?*

Imos precisar tamén unha terceira palabra. Se divides a altura entre a distancia horizontal obtés a **tanxente** do ángulo:

$$\tan \alpha = \frac{\operatorname{sen} \alpha}{\cos \alpha} = \frac{\text{canto sobe}}{\text{canto avanza}}$$

Ten que ver coa *pendente* que indican os sinais de tráfico, que a expresan en porcentaxe: un 6 % quere dicir que por cada 100 metros que se avanza en horizontal sóbense 6, é dicir, unha tanxente de 0,06. A porcentaxe do sinal é a tanxente multiplicada por cen. Unha tanxente igual a 1 (un 100 %) quere dicir que o raio sobe tanto como avanza: é o ángulo de 45°. Unha tanxente igual a 1/2 quere dicir que sobe a metade do que avanza: un ángulo máis deitado, de 26,57°. Canto menor é a tanxente, máis deitado figura o raio.

UN POUCO DE HISTORIA *por que se chama seno?*

O seno naceu na India. Os astrónomos indios, como Aryabhata arredor do ano 500, traballaban coa metade da corda dun arco e chamábanlle jyā ou jīvā, «corda de arco» en sánscrito: debuxada, a corda dunha circunferencia parece a corda dun arco de tirar. Os astrónomos de fala árabe tomaron a palabra tal cal e escribírona jība. Como o árabe escrito adoita omitir as vogais, aquela palabra estranxeira acabou léndose jayb, que en árabe significa «seo» ou «pregue». No século XII, ao pasar estes textos ao latín, tradutores como Xerardo de Cremona traduciron o significado: sinus [92, 116].

Do sánscrito ao galego, pasando polo árabe e o latín: o nome da tecla da túa calculadora é unha cadea de traducións e un erro de lectura.

Formulada así, a pregunta xa non trata de triángulos: trata de *xirar*. E xirar si se pode facer con sumas, se se escolle ben canto se xira.

A idea chámase **CORDIC**, publicouna o enxeñeiro Jack E. Volder en 1959 e é esta. En vez de xirar 37° dunha vez, xírase moitas veces cunha lista fixa de ángulos, cada vez á esquerda ou á dereita, coma quen equilibra unha balanza engadindo pesas nun prato ou no outro. A lista é sempre a mesma, veña o ángulo que veña. O único que cambia é o signo.

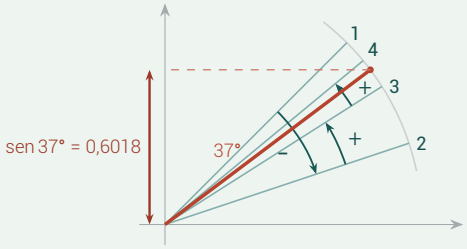
EXEMPLO 1.2 *chegar a 37° a base de xiros*

Empezamos xirando 45°. Despois, en cada paso, miramos onde estamos: se aínda non chegamos a 37°, o seguinte ángulo da lista súmase; se nos pasamos, réstase. Estes son os seis primeiros:

paso	signo	ángulo	onde estamos
1	+	45,0000°	45,0000°
2	−	26,5651°	18,4349°
3	+	14,0362°	32,4712°
4	+	7,1250°	39,5962°
5	−	3,5763°	36,0199°
6	+	1,7899°	37,8098°

Lido en liña: 45 − 26,57 + 14,04 + 7,13 − 3,58 + 1,79 = 37,81. Seis pasos e xa estamos a menos dun grao. As contas fanse cos ángulos enteiros; na

táboa só se ven catro decimais. Se o ángulo buscado fose xustamente 45°, o primeiro xiro daría no cravo e os demais serían idas e voltas arredor del, cada vez máis pequenas. Con quince pasos o erro baixa de tres milésimas de grao; con vinte pasos, baixa de dúas dezmilésimas. Sempre a mesma lista, sempre a mesma operación: sumar ou restar o seguinte da lista.



os catro primeiros xiros: o signo dá o sentido

Pero onde está aquí o seno? Na **altura do punto**. A máquina non leva a conta dos ángulos máis que para decidir o signo de cada xiro: en cada paso actualiza as dúas coordenadas do punto á par, a distancia horizontal e a altura, cunha regra que só usa sumas, restas e metades (o detalle está na sección A.5 do apéndice A).

Con vinte pasos e sempre coa mesma regra (máis unha corrección final, sempre a mesma, porque os xiros estiran un pouco o punto), a altura queda en

0,60181…,

que é o seno de 37° con cinco cifras correctas. Con trinta pasos xa son seis: 0,601815… E de regalo, a distancia horizontal dá o coseno, 0,798636…

» UN CHANZO MÁIS *por que esa lista de ángulos e non outra?*

Queda a pregunta boa, que é a que fai que todo isto funcione: esa lista non é caprichosa. Son ángulos escollidos a propósito **para que xirar por eles non custe ningunha multiplicación**, só sumas e desprazamentos, que é o único que a máquina fai barato.

A explicación está na sección A.5 do apéndice A, coas coordenadas paso a paso, co estiramento que hai que corrixir ao final e coa historia de por que isto se inventou: un bombardeiro no que non cabían as engrenaxes.

1.5 A calculadora aproxima e non o di

Queda unha última cousa e é a máis importante das tres. Fai esta proba nun sitio que amose todas as cifras. O máis a man é a consola do navegador: nun ordenador, abre calquera páxina en Chrome, Edge ou Firefox, preme a tecla F12 (nun Mac, Cmd+Opción+I) e escolle a lapela «Consola». Escribe alí esta suma e preme Intro:

```
0.1 + 0.2
```

Vai co punto e non coa coma porque os programas usan a notación inglesa para os decimais. (Vale igual un programa en Python, se o tes.)

A resposta non é 0,3. É 0,30000000000000004, que na pantalla aparece como `0.30000000000000004`.

Non está avariada e o erro non está na suma: está antes, no momento de gardar o 0,1. A máquina traballa en base dous (capítulo 3) e nesa base o 0,1 é periódico, coma o noso 1/3 en decimal: non se acaba nunca. Así que a máquina non garda 0,1, garda o número máis próximo que pode escribir. E un «case 0,1» máis un «case 0,2» dan un «case 0,3».

> ⚠ **OLLO!** *non é un fallo: é o prezo*
>
> Isto non é un defecto que vaia corrixir a próxima versión. É a consecuencia inevitábel de *intentar* gardar números con infinitos díxitos nun espazo finito [54, 69].
>
> O que si é unha decisión (non unha fatalidade) é que **na pantalla nada o distinga**. Na pantalla, un resultado exacto preséntase exactamente igual ca un aproximado. E o máis curioso é que o aparello si que o *sabe*: nas máquinas que seguen o estándar habitual, por dentro levántase unha marca cada vez que houbo que redondear. O dato existe, pero non chega á pantalla, porque consultalo en cada operación custaría tempo e porque case sempre dá igual.
>
> Queda así a terceira lección do capítulo e a que máis falta nos vai facer: *a máquina aproxima e ningunha parte dela cho di*. Neste caso, porque se decide non mirar a marca. Nos últimos capítulos deste libro, cando a resposta sexa un texto en vez dun número, veremos algo peor: alí **non hai marca ningunha que mirar**.
>
> (Os detalles, o 0,1 en binario e o resto, están no apéndice B.)

1.6 Tres leccións que valen para todo o libro

Mira o que fixemos ata aquí. Multiplicamos sen táboa de multiplicar, calculamos unha raíz que non se pode escribir exacta con cifras e obtivemos un seno sen ningún triángulo. Todo con sumas, restas, metades e algunha división. Da calculadora hai que levar tres cousas e son exactamente as tres que farán falta despois.

1. Procedemento, non coñecemento
Acerta sen saber nada. Non hai dentro ningunha comprensión do que é un seno: hai unha receita que, executada, produce un seno.

2. Moitas pezas e ben encaixadas
Ningún dos pasos é intelixente: sumar, restar, partir pola metade. O comportamento complicado vén da *cantidade* e de *como se relacionan*, non da calidade das pezas.

3. Aproxima e non o di
O resultado é case sempre aproximado e a pantalla é igual de rotunda no caso exacto e no aproximado.

A segunda lección merece un matiz, que se enuncia mellor cunha imaxe: moita area non fai unha casa. Fan falta moitos grans, si, pero sobre todo fai falta unilos con xeito. O que produce o comportamento complicado non é o montón: é o montón *organizado*. Iso vai ser literalmente o argumento dos capítulos 11 e 12.

Hai unha cuarta cousa, máis técnica, que convén levar tamén: nos tres casos apareceu **o dous**. Duplicar e partir pola metade na multiplicación, dividir entre dous na media de Herón, tanxentes $1/2^k$ nos xiros. Non é casualidade: nunha máquina que escribe os números en base dous, multiplicar ou dividir por dous é mover as cifras un lugar, a operación máis barata que hai: faise nun só golpe de reloxo e con moi poucos circuítos, así que sae barata dúas veces, en tempo e en enerxía. Os algoritmos que sobreviviron dentro das máquinas son, en boa medida, os que souberon apoiarse niso.

A terceira lección é a que máis nos importa. Cando máis adiante vexamos que un programa de intelixencia artificial escribe unha data falsa coa mesma seguridade ca unha verdadeira, non haberá que buscar explicacións psicolóxicas nin falar de mentiras. Estará pasando o mesmo que aquí, pero no texto escrito pola máquina, moito máis complexo: **un procedemento que produce a saída máis plausíbel**

segundo as súas instrucións, sen ningún mecanismo interno que distinga «isto é exacto» de «isto é un apaño». Aquí, na calculadora, a marca existe e decídese non mirala. Alí, peor aínda, non a hai. E ademais entran dous ingredientes novos: unha pinga de azar posta a propósito (capítulo 13) e uns datos de partida que ninguén garante que fosen certos (capítulo 15).

SEN MEDO *se leches ata aquí*

Xa entendiches a parte conceptualmente máis difícil do libro, aínda que non o pareza.

Todo o resto (os bits, as palabras convertidas en puntos, o adestramento, a atención) son estas tres ideas outra vez, con máis pezas e máis capas. Se nalgún momento máis adiante te perdes, volve a esta páxina: case sempre a confusión vén de ter esquecido que dentro só hai unha receita longa.

O ESENCIAL DESTE CAPÍTULO

- Unha calculadora non ten as respostas gardadas: **prodúceas** executando un algoritmo con moi poucas operacións básicas.
- Multiplicar pódese facer duplicando e sumando, porque todo número se descompón en suma de potencias de dous. Esta é a idea que sostén todo o capítulo 3.
- A raíz cadrada calcúlase **estimando e corrixindo**: pártese dunha estimación mala, mídese o erro e corríxese a estimación. Cada volta duplica as cifras correctas. Esa é, tal cal, a idea do capítulo 10.
- O seno obtense **xirando**, non medindo, cunha lista curta de ángulos escollidos para que o xiro só custe sumas e metades.
- O que se garda dentro non son respostas: son ferramentas. Vinte números, non oitenta e un millóns.
- Case todos os resultados son **aproximados e na pantalla nada os distingue** dun exacto. Esta é a lección que máis falta vai facer despois.
- As receitas completas, incluída a da raíz cadrada que se aprendía na escola, co seu debuxo, están no apéndice A.

E AGORA?

Acabamos de explicar como funciona unha máquina. Pero que significa exactamente «explicar como funciona»? Antes de subir ao seguinte chanzo convén afiar esa pregunta, porque de aquí en diante vai ser moi fácil confundir describir un comportamento con explicar un mecanismo. Diso vai o capítulo 2.

Capítulo 02

Que quere dicir «saber como funciona»?

«O que non podo crear, non o entendo.»
Richard Feynman, atopado no seu taboleiro en 1988

PARA ARRANCAR

Sabes como funciona unha bicicleta? Case todo o mundo di que si. Pero cando se lle pide á xente que complete o debuxo dunha bicicleta, unha parte importante debuxa unha que non podería moverse. Que é, entón, o que sabemos cando dicimos que sabemos como funciona algo?

2.1 A ilusión de entender

No capítulo anterior abrimos unha calculadora. Antes de abrir cousas máis grandes convén parar un momento e preguntarse que estamos a facer exactamente cando dicimos que entendemos como funciona unha máquina. Non é unha pregunta filosófica de adorno: se non a respondemos, dentro de dez capítulos non saberemos se entendemos algo ou se só nos quedamos coa sensación.

Porque a sensación engana e faino de maneira mediбel.

UN POUCO DE HISTORIA *o experimento das bicicletas*

En 2006, a psicóloga Rebecca Lawson pediulles a estudantes universitarios que puntuasen canto crían que entendían o funcionamento dunha bicicleta. A maioría púxose unha nota alta: unha bicicleta é un obxecto cotián, sen misterio.

Despois deulles un debuxo incompleto (o cadro sen algunhas barras, sen cadea, sen pedais) e pediulles que o completasen. Unha proporción grande debuxou bicicletas imposíbeis: a cadea unindo as dúas rodas (co cal a bicicleta non podería xirar), ou barras do cadro atravesando o espazo por onde ten que pasar a roda [84]. E o máis interesante: moitas persoas seguían sen ver o problema despois de debuxalo.

Uns anos antes, Leonid Rozenblit e Frank Keil puxéranlle nome a este efecto. Chamáronlle ***ilusión de profundidade explicativa****: tendemos a crer que entendemos os mecanismos moito mellor do que os entendemos e a ilusión desfaise en canto nos piden que os expliquemos paso a paso [131].*

E hai unha versión do experimento que se pode mirar, que é o mellor que lle pode pasar a unha idea coma esta. O deseñador italiano Gianluca Gimini levaba desde 2009 pedíndolle á xente que lle debuxase de memoria, alí mesmo, unha bicicleta. Xuntou 376, de persoas de tres a oitenta e oito anos. Despois colleu as máis chamativas e fíxoas en tres dimensións, coas cores e os reflexos dun catálogo, tal e como estaban debuxadas [53]. O resultado é unha colección de máquinas que non se sosteñen, non xiran ou non avanzan e que son, á vez, moi fermosas e moi divertidas. Gimini contou despois que el empezara aquilo por gusto e que só máis tarde soubo que lles estaba a facer a persoas descoñecidas unha proba que en Psicoloxía se usa desde hai anos.

Hai un detalle do experimento que é o máis útil para nós. A ilusión non se desfai lendo máis. Desfaise **ao intentar producir a explicación**. E esta é unha propiedade curiosa: a única proba fiábel de que entendes algo é poder explicarllo a alguén que non o sabe, en voz alta, sen mirar.

PÉNSAO UN MOMENTO *proba rápida*

Antes de seguir, escolle un destes tres e intenta explicalo en voz alta, agora, durante trinta segundos, coma se llo contases a unha persoa de doce anos: *como se abre unha pechadura cunha chave*; *por que flota un barco de ferro*; *que pasa cando premes o interruptor da luz*.

Se che saíu fluído, perfecto. Se notaches que a explicación se che desfacía no medio da frase, acabas de experimentar en directo o efecto do que fala esta sección. Non é un problema teu: pásalle a todo o mundo e é reparábel.

SEN MEDO *isto é unha boa noticia*

Pode soar a reproche e é o contrario. O que dixemos ata aquí significa que **a sensación de non entender de todo é un sinal de honestidade, non de torpeza**.

Se durante este libro chegas a un punto no que pensas «isto crin que o entendía e resulta que non», non retrocedas: estás mellor ca antes. Quen nunca chega a ese punto é porque se quedou na sensación.

2.2 Tres preguntas que non son a mesma

«Como funciona» é unha pregunta ambigua. En realidade son tres e confundilas é a causa de case todas as discusións estériles sobre intelixencia artificial.

O neurocientífico David Marr, estudando a visión, propuxo distinguilas con claridade [93]. Aplicámolas á calculadora do capítulo anterior:

1. Que fai? *o problema*

Dá o seno dun ángulo. Non se fala aínda de como. Só de cal é a tarefa.

2. Como o fai? *o algoritmo*

Xirando vinte veces cunha lista fixa de ángulos e escollendo cada vez o signo. Aquí é onde vive este libro.

3. Con que o fai? *a realización*

Transistores de silicio que deixan pasar ou non deixan pasar corrente. Poderían empregarse outras cousas: engrenaxes, auga ou fichas de dominó.

Hai unha cuarta pregunta que convén non mesturar coas tres: «*Para que o fai?*». Esta pregunta non se refire ao mecanismo, senón á finalidade. A súa resposta non está dentro da máquina: está en quen a manda facer. Dicímolo en presente a propósito. *Non está* é o que se pode comprobar abrindo a máquina de hoxe; *nunca estará* sería unha profecía e neste libro intentamos non facer ningunha.

O importante é que **as tres son independentes**. Podes coñecer unha e ignorar as outras dúas sen contradición ningunha.

Un exemplo doméstico: sabes perfectamente *qué fai* un reloxo (nivel 1) e probabelmente non saibas *como* o fai (nivel 2: un cristal de cuarzo que vibra un número fixo de veces por segundo e un contador que as conta). E case seguro que non sabes nada do nivel 3.

OLLO! *a confusión máis común sobre a intelixencia artificial*

Case todas as discusións atascadas sobre estas máquinas veñen de mesturar os niveis. Dúas mostras:

«*É que por dentro son redes neuronais, igual có cerebro.*» Isto pretende ser unha explicación de nivel 3 e non o é: as neuronas artificiais non se parecen ás biolóxicas máis do que un avión se parece a unha gaivota: o que hai dentro son táboas de números e operacións aritméticas con elas. Volveremos a isto no capítulo 11.

«*É que estas máquinas só predín a palabra seguinte.*» Isto é certo e é de nivel 1: describe a tarefa. Pero preséntase coma se fose unha explicación completa e non explica nada do nivel 2, que é onde está a parte máis interesante. Dicir que unha bicicleta «só transmite o movemento das pernas á roda» tamén é certo e tampouco che permite construír unha.

EN PALABRAS SINXELAS *en que nivel está escrito este libro?*

No segundo, case sempre.

Non imos falar de silicio nin de tarxetas gráficas máis do imprescindíbel (nivel 3) e non nos imos conformar con dicir que o programa «predí texto» (nivel 1). O obxectivo é que ao rematar poidas contar, paso a paso, *que operacións se executan* entre a túa pregunta e a resposta. Ese é o nivel no que a palabra «entender» significa algo comprobábel.

2.3 Que é exactamente un algoritmo

Usamos a palabra no capítulo anterior sen definila, á beira doutra máis caseira que vén sendo o mesmo: *receita*. Convén facelo ben, porque é o concepto central de todo o libro.

Definición 2.1 *Algoritmo* Para os propósitos deste libro, un **algoritmo** é un procedemento ben definido que toma uns **datos de entrada** e, nun tempo finito, produce unha **saída**: unha serie de pasos que transforma o que entra no que sae [33]. Pedímoslle catro condicións:

1. Consta dun número **finito** de instrucións.
2. Cada instrución é **inequívoca**: dous executores distintos interprétana do mesmo xeito.
3. Cada instrución ten **efecto finito**: cambia unha cantidade acoutada de cousas e faino nun tempo acoutado.
4. Con calquera entrada válida, **remata sempre** e a saída é a que se pedía.

A terceira condición parece un tecnicismo e non o é. «Súmalle un», «bórrao», «cambia esta letra pola seguinte» valen: cada unha toca un anaco pequeno e sábese cal. «Namórate» non vale e non por ser vaga. Aínda que dúas persoas entendesen neste caso exactamente o mesmo, non hai un cambio concreto e acoutado que executar. Esa é a condición que permite *contar* os pasos dun procedemento e polo tanto medir canto custa, que é do que vai boa parte deste libro.

A cuarta parece obvia e non o é tanto. Un procedemento que, con algunha entrada, se queda dando voltas para sempre non cumpre a súa parte do trato, que era darche a saída. Por iso os algoritmos levan unha *condición de parada* explícita: «ata que dúas voltas seguidas coincidan», «ata que non queden elementos». Sen ela non hai saída garantida e, polo tanto, non hai algoritmo.

As catro condicións son restritivas de verdade, aínda que menos do que parece. Unha receita de cociña pode ser perfectamente un algoritmo: «bátese ata que estea no punto» é unha condición de parada coma calquera outra, sempre que *o punto* estea definido de maneira que dúas persoas distintas o recoñezan do mesmo xeito (por exemplo: «ata que a mestura forme picos que non caen»). O que arruína a condición segunda non é a cociña: é a vaguidade. «Sal ao gusto» non se pode executar sen decidir por conta propia.

Iso é o que distingue as instrucións que lle damos a unha persoa das que lle damos a unha máquina: ás persoas dáselles marxe porque teñen criterio para encher os ocos. A unha máquina hai que pecharllos todos, incluído o de cando parar. E de aí vén unha pregunta que ten unha resposta sorprendente ao final desta sección: *pódese saber de antemán se un programa calquera vai parar?* Fíxate en que a pregunta non é sobre algoritmos, que por definición rematan, senón sobre programas escritos por calquera: a dúbida é xustamente se ese programa é un algoritmo ou non.

UN CHANZO MÁIS *cando a saída non chega ou non pode chegar*

A definición de arriba é a práctica, a dos manuais de algoritmos. Abonda para todo o libro, pero non describe todo o mundo dos procedementos e convén sabelo. Hai procedementos ben escritos que con algunhas entradas dan resposta e con outras quedan dando voltas para sempre; o que calculan chámase *función parcial*, porque só está definida para parte das entradas. E hai problemas para os que non existe ningún algoritmo, por moito que se busque. O mesmo manual lembra que, con eles, calquera procedemento que se intente falla dun destes dous xeitos: ou responde sempre pero ás veces mal, ou cando responde acerta pero con algunhas entradas non responde nunca [33]. Chámanse problemas *non computábeis* e o exemplo máis famoso aparece ao final desta sección.

POR QUE FUNCIONA? *a propiedade que fai posíbel todo o demais*

Da definición dedúcese algo que parece menor e é o fundamento de toda a informática:

> **Un algoritmo pode ser executado correctamente por alguén que non entende para que serve**

Se as instrucións son inequívocas, quen as segue non precisa saber nada máis. Pode executar á perfección o método de Herón sen ter idea de que é unha raíz cadrada, do mesmo xeito que se pode seguir unha partitura sen saber que melodía sairá.

Isto ten unha consecuencia que vai doer máis adiante: **producir a saída correcta non demostra comprensión**. Non a demostra nunha máquina e tampouco, por certo, nun exame.

UN POUCO DE HISTORIA *unha palabra que é un nome propio*

A palabra algoritmo vén do nome dunha persoa. Muhammad ibn Musa al-Juarismi foi un sabio que traballou en Bagdad no século IX e escribiu un tratado sobre o cálculo cos numerais indios. Ao traducirse ao latín na Europa medieval, o seu nome deformouse en algorismus e acabou designando primeiro o cálculo con cifras arábigas e despois calquera procedemento mecánico de cálculo.

Doutra obra súa, sobre a resolución de ecuacións, sae unha segunda palabra que tamén usamos todos os días: al-yabr, que deu álxebra [70].

UN CHANZO MÁIS *hai cousas que ningún algoritmo pode facer*

Se un algoritmo é un procedemento mecánico, a pregunta natural é: *hai problemas para os que non existe ningún?*

A resposta é si e demostrárona en 1936, case á vez e por camiños distintos, Alonzo Church e Alan Turing, antes de que existise ningún ordenador [28, 153]. Para poder demostrala, Turing tivo que definir primeiro, con precisión matemática, que é «un procedemento mecánico». Inventou unha máquina imaxinaria moi simple (unha cinta, un cabezal que le e escribe símbolos e unhas regras) e defendeu que calquera cálculo mecánico se pode facer con ela. Iso último non se pode demostrar, porque «cálculo mecánico» non é unha idea matemática precisa; chámaselle **tese de Church-Turing**. Pero cada definición distinta que se propuxo desde aquela, empezando pola de Alonzo Church no mesmo ano [28], resultou permitir exactamente os mesmos cálculos. Por iso case todo o mundo a dá por boa.

Despois demostrou que hai preguntas que esa máquina non pode responder nunca. A máis famosa: *dado un programa calquera e a súa entrada, decidir se ese programa acabará algún día ou quedará dando voltas para sempre*. Non é que non se saiba facer: está demostrado que non se pode.

Convén enunciar este resultado enteiro, coas súas dúas metades, porque a metade que se esquece é a importante. Non se trata de que a pregunta sexa difícil nin de que a nós non se nos ocorra a resposta. O que Turing demostrou é que **non existe ningún procedemento mecánico que o decida para todos os casos**: ningunha máquina das súas pode examinar outra máquina das súas e responder sempre ben. O que non se pode é *mecanizar*

a pregunta. Esa é unha metade. A outra: no caso dun programa concreto, moitas veces abonda con miralo por riba para sabelo.

Volveremos a este resultado no capítulo 13, cando toque preguntar como sabe unha máquina destas que xa rematou de responder (a resposta é menos tranquilizadora do que parece). E convén sabelo, ademais, porque de cando en vez alguén anuncia que unha intelixencia artificial resolverá calquera problema. Hai problemas que unha máquina non pode resolver, por grande que sexa. Por iso, non ten sentido agardar que o faga. Non é unha cuestión de potencia.

2.4 Dentro non hai o que hai fóra

Agora vén a idea máis difícil de aceptar deste capítulo e a que máis se resiste, porque vai contra unha intuición moi forte.

Un programa que xoga moi ben ao xadrez non contén ningún taboleiro; contén números e comparacións. Un programa que predí o tempo non contén nubes nin humidade; contén táboas de números que se actualizan segundo unhas ecuacións. A calculadora do capítulo anterior non contén ángulos coma os do transportador; contén unha lista de vinte números que os representan e unha regra para sumalos.

MÍRAO *o mapa e o territorio*

Un mapa de estradas non contén asfalto. Contén liñas sobre papel que *gardan certas relacións* coas estradas: se dúas cidades están preto no territorio, están preto no mapa.

Esa correspondencia é o que fai útil o mapa e é tamén o que ten de enganoso. O mapa non ten pendentes, nin curvas pechadas, nin obras. Un mapa recolle *algunhas* relacións e tira o resto e o que tirou non está en ningún sitio do mapa esperando a ser atopado.

Cando digamos, no capítulo 7, que un modelo ten «o significado das palabras» convertido en distancias, convén ter esta imaxe diante. Hai un mapa. Recolle relacións reais. E hai cousas que non están nel de ningunha maneira.

As palabras prestadas

Todo o vocabulario que usamos para falar destas máquinas está tomado em-

prestado dos seres humanos. Non foi unha conspiración: foi comodidade e en parte entusiasmo. Pero o efecto acumulado é considerábel, porque as metáforas non son neutras: traen con elas todo o que sabemos do orixinal.

Palabra	O que suxire	O que designa aquí
aprender	Comprender algo novo, cambiar de idea, darse conta.	Axustar millóns de números para reducir un erro medido. Cap. 10.
neurona	Unha célula viva do cerebro.	Unha suma ponderada seguida dunha regra moi simple. Cap. 11.
atención	Concentrarse voluntariamente nunha cousa.	Un cálculo de pesos que decide canto pesa cada palabra anterior. Cap. 12.
alucinar	Percibir algo que non existe, por unha alteración.	Producir a continuación máis plausíbel cando non hai unha correcta. Cap. 14.
entender	Captar o sentido.	Non hai equivalente. Esta é a palabra que máis convén non usar.
memoria	Lembrar o vivido.	Un espazo onde caben as últimas palabras do diálogo e nada máis. Caps. 8 e 13.

Non propoñemos abolir estas palabras: sería inútil e pedante e, ademais, son cómodas. Propoñemos outra cousa, máis modesta: que cando as leas *marcadas así* te lembres, medio segundo, de que estás usando unha comparación. Medio segundo abonda.

PÉNSAO UN MOMENTO *por onde sae o sol?*

Hai un exemplo moito máis vello deste fenómeno e está tan metido na lingua que ninguén o nota.

Dicimos que o sol *sae* polo leste e *se pon* polo oeste. Dise así todos os días, tamén entre xente que sabe perfectamente que o sol non sae de ningunha parte: é a Terra a que xira e nos vai poñendo diante del. A frase é unha reliquia de cando a humanidade pensaba que a Terra estaba quieta no centro e era o ceo o que daba voltas e sobreviviu a Copérnico e a Galileo

sen despeitearse.

Poderíase pensar que é inofensiva, porque todo o mundo sabe o que quere dicir. Pero cada xeración nova apréndea antes de aprender a astronomía e a imaxe que deixa (un sol que se move e unha Terra quieta) é a que primeiro se instala. Nunha enquisa da National Science Foundation estadounidense de 2014, un 26 % das persoas adultas, unha de cada catro, respondeu que é o Sol o que xira arredor da Terra [105]. A palabra non só describe mal: **ensina mal** e faino antes ca ninguén.

Con *aprender*, *entender* ou *alucinar* pode pasar o mesmo. Unha xeración que medre dicindo que a máquina «entende» vai ter moi instalada a idea de que entende antes de preguntarse que hai dentro. Por iso as marcamos.

O QUE SE DI POR AÍ *«se se comporta coma se entendese, é que entende»*

É unha posición respectábel, con notables partidarios e partidarias: se unha máquina resolve todos os exames que lle poñamos, dicir que «non entende de verdade» pode soar a un truco para non recoñecer o evidente. Alan Turing argumentou algo parecido en 1950 cando propuxo substituír a pregunta «poden pensar as máquinas?» (que dá título, por certo, a un libro en galego de Senén Barro [13]) por outra máis manexábel, baseada só no comportamento [154]: *poden imitar?* Turing considerou a pregunta orixinal mal definida para discutila e propuxo cambiala por outra que se pode comprobar.

Hai obxeccións igualmente serias. A máis coñecida é o experimento mental de John Searle: unha persoa pechada nun cuarto, que non sabe chinés, recibe papeis con caracteres chineses e ten un manual enorme que lle di que caracteres devolver. Desde fóra, as respostas son perfectas. Searle sostén que esa persoa non entende chinés en ningún momento e que engadirlle velocidade non cambia nada [136].

Non imos resolver aquí unha discusión que leva máis de corenta anos aberta; nin sequera temos claro que sexa resolúbel tal e como está formulada. Quen queira meterse nela ten un clásico enteiro dedicado a estas voltas [67]. O que si podemos dicir é o que nos parece máis útil: **a discusión é máis produtiva cando se sabe que hai dentro do cuarto**. Por iso este libro describe o manual con detalle en vez de opinar sobre a persoa.

2.5 ELIZA e o xogo da imitación

Turing chamoulle *xogo da imitación* ao que propuxo en 1950: se unha máquina consegue responder de tal xeito que quen conversa con ela non é quen de distinguir se do outro lado hai unha persoa ou un programa, gaña o xogo. Fíxate ben no que pide o xogo, porque non é o que moitas veces se conta. Non pide que a máquina pense. Pide que *imite* ben abondo. Turing apostaba a que o segundo era alcanzábel e deixaba o primeiro sen resolver; ese é exactamente o punto interesante.

Dezaseis anos despois, un programa enganou a moita xente sen pretendelo e cunhas poucas regras.

Joseph Weizenbaum, no MIT (o Instituto Tecnolóxico de Massachusetts, polas súas siglas en inglés), escribiu un programa chamado ELIZA. Un dos seus guións, o chamado DOCTOR, imitaba o estilo dun psicoterapeuta rogeriano, que é un estilo que se presta moito: devolve as preguntas, non achega información propia e pide que se elabore [165].

O programa non tiña ningún modelo da conversa, nin memoria, nin coñecemento do mundo. Tiña unha lista de *patróns* e, para cada un, unha resposta. Nada máis. Aquí tes unhas regras do mesmo tipo, en galego, simplificadas ata o óso:

	Se a frase...	**entón responde**
R1	contén «sempre»	«Podes darme un exemplo concreto?»
R2	empeza por «eu son»	«Canto tempo levas sendo ... ?»
R3	contén «o meu» ou «a miña»	«Cóntame máis sobre o teu ou a túa ... »
R4	contén «porque»	«É esa a verdadeira razón?»
R5	contén «non»	«Por que non?»
R6	(ningunha das anteriores)	«Segue, por favor.»

Aplícanse por orde: gaña a primeira que encaixe. Vexamos qué sae.

EXEMPLO 2.1 *unha conversa con seis regras*

Persoa	Programa	Regra
A miña irmá non me fala.	Cóntame máis sobre a túa irmá.	R3
Sempre foi así.	Podes darme un exemplo concreto?	R1
Non quero falar diso.	Por que non?	R5
Porque me poño triste.	É esa a verdadeira razón?	R4
Eu son unha persoa reservada.	Canto tempo levas sendo unha persoa reservada?	R2

Le a columna do programa tapando a da persoa e a das regras. Parece alguén que escoita con atención: recolle o que dis, pide concreción, insiste con delicadeza. Agora le só a columna das regras. Non hai ninguén aí.

PRÓBAO TI 2.1 ★☆☆

Aplica as seis regras, por orde, a estas tres frases e escribe que respondería o programa:

1. «O meu xefe non me valora.»
2. «Eu son bastante desconfiado.»
3. «Onte choveu moito.»

Despois fai o interesante: **escribe unha frase que rompa o programa**, é dicir, unha na que a resposta quede claramente absurda. Que tipo de frase o consegue? Que che di iso sobre o que o programa non ten?

resposta na páx. 367 →

Weizenbaum non pretendía enganar a ninguén: escribiu ELIZA para amosar precisamente o superficial que era a comunicación entre persoa e máquina. Pasou o contrario. A xente estableceu conversas longas, pedía quedar a soas co terminal e psiquiatras profesionais chegaron a suxerir en serio que un sistema así podería atender pacientes. Weizenbaum quedou tan alarmado que dedicou boa parte do resto da súa vida a escribir contra o uso acrítico dos ordenadores [166]. Ao efecto de atribuírlle comprensión a un programa polo simple feito de que as súas respostas encaixen, chámaselle desde entón **efecto ELIZA**.

UN POUCO DE HISTORIA *cando ELIZA conversou cun paciente que tampouco existía*

A historia ten unha segunda parte menos coñecida. En 1972, o psiquiatra Kenneth Colby escribiu en Stanford un programa chamado PARRY que facía o contrario de ELIZA: en vez de imitar a quen escoita, imitaba a quen fala [31]. Concretamente, a un paciente con esquizofrenia paranoide, coas súas obsesións fixas e cun nivel de desconfianza que subía ou baixaba segundo o que lle dixesen.

O 18 de setembro dese ano puxéronos a conversar entre eles a través da ARPANET, a rede dos Estados Unidos da que sairía a internet: PARRY corría na Universidade de Stanford, en California; ELIZA, na outra costa, en Cambridge (Massachusetts), nun ordenador de Bolt, Beranek e Newman, unha empresa fundada por profesores do MIT. A transcrición publicouse como un dos documentos técnicos cos que se foi construíndo aquela rede [26] e é o que cabería esperar: dous programas sen nada dentro dándolle voltas a unha conversa que non existe.

O dato que importa aquí é outro. Colby fixo a proba en serio. Psiquiatras de verdade entrevistaron a PARRY por teletipo sen que se lles avisase de que podía haber unha máquina do outro lado. Despois, outro grupo de psiquiatras leu esas transcricións mesturadas coas de pacientes reais e tratou de dicir cal era a do programa. Acertaron o 48 % das veces [135], que é o mesmo que daría tirar unha moeda, é dicir, decidir ao chou sen ler nada.

Non é que PARRY entendese nada. É que, nese formato e sobre ese tema, conversar non servía para distinguir.

Ademais, convén non esaxerar o resultado. A PARRY tocoulle un papel moi doado de imitar: un programa con poucos recursos responde con evasivas e cambia de tema: xusto o que se espera dun paciente paranoide. Os psiquiatras non quedaron sen saber distinguir porque o programa fose bo, senón porque os seus defectos parecían síntomas.

OLLO! *o que NON demostra este exemplo*

Aquí cómpre precisión, porque este exemplo se usa moitas veces mal.

Non demostra que os modelos actuais sexan «ELIZA con máis memoria». Non o son e a diferenza non é de grao: ELIZA tiña unhas decenas de regras escritas a man, non extraía ningunha estrutura do idioma e non podía responder nada que non estivese previsto polo seu autor. Un modelo actual constrúe as súas respostas a partir de regularidades que ninguén lle escribiu e resolve tarefas que ELIZA nin podía formular. Comparalos como iguais é faltar á verdade.

O que **si** demostra ELIZA e segue valendo enteiro, é algo máis modesto e máis útil: que a nosa capacidade de inferir comprensión a partir da conversa é *moi* pouco fiábel. Xa en 1966, con corenta regras, bastaba para convencer a xente intelixente. Isto non nos di nada sobre o que hai dentro dun modelo moderno. Dinos moito sobre o pouco que nos podemos fiar da nosa impresión ao lelo e, polo tanto, sobre por que fai falta mirar dentro.

2.6 Que imos entender e que non

Xa se pode dicir con precisión o que promete o resto do libro.

Vas poder contar o percorrido completo. Ao rematar o capítulo 13 serás quen de explicar, con palabras túas e sen mirar, que lle pasa á túa pregunta desde que premes «enviar»: como se corta en anacos, como se converte cada anaco nun punto dun espazo, qué fan as capas sucesivas, como sae unha lista de candidatos con probabilidades, como se escolle un e por que non sempre o mesmo.

Vas ter criterio. Vas saber por que estas máquinas son extremadamente boas nunhas cousas e sistematicamente malas noutras e vas poder predicir en cal dos dous casos estás antes de fiarte.

Non vas poder construír unha e ningún libro coma este che vai dar iso.

E hai un límite máis serio, que convén deixar dito porque non é unha limitación deste libro senón do estado actual do coñecemento:

EN PALABRAS SINXELAS *o mecanismo si; a explicación de cada resposta, non*

Distingamos dúas preguntas que se confunden a miúdo:

Como funciona un modelo de linguaxe? Sábese enteiro. Está publicado, descrito operación a operación e calquera con recursos suficientes pode construír un. É o contido deste libro.

*Por que **este** modelo respondeu **iso** a **esta** pregunta?* Non se sabe, agás en casos moi simples. Non por segredismo: porque a resposta depende de centos de milleiros de millóns de números axustados automaticamente e ler neles unha explicación é un problema aberto no que traballa unha disciplina enteira.

A analoxía honesta non é «unha caixa negra que ninguén entende». É máis parecido á meteoroloxía: coñecemos as ecuacións e o mecanismo das nubes con todo detalle e aínda así non podemos dicir a que hora exacta vai chover no teu barrio dentro de dez días. Coñecer o mecanismo e poder predicir un caso concreto son cousas distintas e a primeira, por si soa, xa é moitísimo.

PARA QUE? *por que paga a pena esta distinción?*

Porque é a que che vai permitir detectar as dúas clases de mentira que circulan sobre este tema.

Cando alguén diga «isto é unha caixa negra, non se pode saber nada», saberás que está mesturando as dúas preguntas: o mecanismo si se sabe e negalo adoita servir para evitar dar explicacións sobre decisións que si se poderían explicar.

E cando alguén diga «sabemos exactamente o que fai en cada momento», saberás que está esaxerando na dirección contraria, normalmente para vender algo.

Ter clara esta distinción (mecanismo coñecido fronte a caso concreto opaco) é, probabelmente, a ferramenta máis útil que che vai deixar este libro.

PÉNSAO UN MOMENTO *e nós, que facemos cando entendemos?*

Levamos o capítulo enteiro dicindo que a máquina non entende. Xusto é devolverlle a pregunta a quen a fai: *que facemos nós cando dicimos que entendemos algo?*

Pola proba do principio do capítulo sabemos polo menos unha cousa: non é ter a sensación de entender. A bicicleta demostrouno. O que si parece distinguir o entender humano é poder **producir a explicación**, responder preguntas novas sobre ela, detectar cando non encaixa e dicir «espera, isto non me cadra».

Non hai acordo científico sobre qué é exactamente iso nin sobre cómo o fai un cerebro. Pero si podemos usalo como criterio práctico, o mesmo para as persoas e para as máquinas: quen entende algo pode explicalo, aplicalo a un caso novo e recoñecer cando falla. Ao longo do libro iremos vendo cales desas tres cousas fai unha máquina de escribir texto e cales non.

O ESENCIAL DESTE CAPÍTULO

- Crer que entendemos un mecanismo é moi distinto de entendelo: a **ilusión de profundidade explicativa** desfaise ao intentar explicar paso a paso, non ao ler máis.
- «Como funciona» son tres preguntas: **que fai**, **con que procedemento** e **de que está feito**. Este libro vive na segunda.
- Un **algoritmo** é un procedemento finito e inequívoco e a súa propiedade decisiva é que se pode executar sen entendelo. Por iso producir a saída correcta non demostra comprensión.
- **Dentro non está o que hai fóra**: hai un mapa que garda algunhas relacións e tira o resto.
- Todo o vocabulario (*aprender*, *atención*, *alucinar*) está tomado emprestado de nós. Son metáforas útiles, non descricións.
- **ELIZA** amosou en 1966 que a nosa impresión ao conversar non serve para deducir o que hai dentro. Isto non equipara ELIZA cos modelos actuais: só descualifica a nosa intuición como proba.
- O mecanismo xeral **sábese**; a razón dunha resposta concreta, non. Son

dúas cousas distintas e confundilas é a fonte de case todo o ruído nos debates sobre intelixencia artificial.

E AGORA?

Dixemos que dentro só hai números e que todo o demais son receitas. Pero se dentro só hai números, hai un problema evidente: as palabras non son números, as fotos non son números e a música tampouco. Como entrou aí todo iso? A resposta empeza por contar con dous dedos.

BLOQUE I

Todo é número

Dentro non hai letras, nin cores, nin son, nin sentido. Hai números. Como chegou aí todo o demais e que se perde polo camiño?



CAPÍTULOS 3 A 5

Capítulo 03

Contar con dous dedos

PARA ARRANCAR

Por que unha máquina conta con ceros e uns e non con dez cifras coma nós? Non foi por capricho nin por afán de complicar a vida: é a maneira máis segura de contar, a que menos erros deixa pasar. E é a física a que explica por que é moito máis doado conseguir esa seguridade con dous estados eléctricos ca con dez.

3.1 Dez dedos, dez cifras

Escribe 3 527. Líchelo sen pensar e con razón: levas facéndoo desde os seis anos. Pero mira o que acabas de dar por suposto.

Hai catro símbolos. O 3 non significa tres: significa *tres milleiros*. O 5 non significa cinco: significa *cincocentos*. O mesmo símbolo vale cousas distintas segundo onde estea:

$$3\,527 = 3 \times 1000 + 5 \times 100 + 2 \times 10 + 7$$

A isto chámaselle **valor posicional** e é un dos inventos máis potentes da historia. Non o fixo ninguén dunha vez: os babilonios xa o usaban en base sesenta; na China da dinastía Han facíase en base dez con varillas sobre un taboleiro, deixando un oco baleiro onde nós poñemos un cero; os maias de Mesoamérica facíano en

base vinte, cun signo propio para o cero [70]; e as cifras que usamos hoxe, co cero incluído, chegaron da India a través do mundo árabe, como vimos no capítulo 2. Sen el, escribir números grandes é un pesadelo: proba a multiplicar MMMDXXVII por LIV en números romanos e verás. Con el, catro cifras abondan para calquera cantidade por debaixo de dez mil.

Fíxate agora nos números que aparecen nesa suma: 1000, 100, 10, 1. Son as potencias de dez. E a pregunta que case ninguén fai é: *por que dez?*

EN PALABRAS SINXELAS *o dez non ten nada de especial*

Non hai ningunha razón matemática para escoller o dez. Ningunha. O dez non ten propiedades que o fagan mellor có oito, o doce ou o dezaseis; de feito, para repartir, como veremos axiña, é bastante malo.

Unha explicación é anatómica: temos dez dedos. Contamos con eles antes de contar con símbolos e o sistema de escritura quedou casado con esa casualidade biolóxica. Se a nosa especie tivese oito dedos, escribiriamos os números doutra maneira e a aritmética funcionaría exactamente igual de ben.

UN POUCO DE HISTORIA *os babilonios contaban de sesenta en sesenta*

Hai case catro mil anos, en Mesopotamia, usábase un sistema de base sesenta. Parece unha extravagancia ata que se mira o motivo: sesenta divídese moi ben.

Os divisores de 60 *son* 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 *e* 60*: doce en total. Os de* 10 *son só catro:* 1, 2, 5, 10*. Nunha cultura que reparte terras, cereais e herdanzas sen decimais, poder dividir sen resto entre dous, tres, catro, cinco e seis é unha vantaxe enorme. E os de* 12 *son seis, que é unha das razóns das ducias e das dúas ducias de horas.*

Aquel sistema deixou unha pegada que aínda usamos sen pensar nela: os sesenta minutos dunha hora, os sesenta segundos dun minuto e os trescentos sesenta graos dunha circunferencia (seis veces sesenta). Chegaron ata nós a través da astronomía grega, que calculaba con fraccións en base sesenta herdadas de Mesopotamia. O costume de medir a vida diaria en minutos e segundos é, en cambio, moito máis recente.

Non foi a única cultura que lle viu a vantaxe ao sesenta. Na China, polo

menos desde o segundo milenio antes da nosa era (xa aparece nos ósos oraculares da dinastía Shang), os días e despois tamén os anos noméanse cun ciclo de sesenta. Combínanse dúas series, unha de dez nomes e outra de doce, e cada día leva un de cada. Por que sesenta e non cento vinte? Porque sesenta é o primeiro número que é múltiplo de dez e de doce á vez: despois de sesenta pasos as dúas series volven coincidir no principio. O ciclo actual empezou en 1984 e o seguinte empezará en 2044 [78].

PÉNSAO UN MOMENTO *oito dedos*

Imaxina unha especie coma nós pero con catro dedos en cada man. Contarían de oito en oito e para eles o número que escribimos 8 escribiríase 10.

Pregunta: *as súas matemáticas serían distintas das nosas?* Sería distinto o que é un número primo? Sería distinto o teorema de Pitágoras? Sería distinta a raíz cadrada de dous?

A resposta importa para todo o capítulo.

3.2 Contar cando só hai dúas opcións

Imos construír agora un sistema de numeración con dous símbolos (0 e 1) e nada máis. A regra é a mesma ca sempre: cando se acaban os símbolos dunha columna, ponse a cero e engádese un á columna da esquerda. Só que aquí os símbolos acábanse moito antes.

decimal	binario	decimal	binario
0	`0`	5	`101`
1	`1`	6	`110`
2	`10`	7	`111`
3	`11`	8	`1000`
4	`100`	9	`1001`

Pensa nun contaquilómetros antigo, deses de rodiñas. No noso, cada rodiña ten dez posicións e, ao pasar do 9, volve ao 0 e empurra a seguinte. Aquí cada rodiña ten dúas: 0 e 1. Vólvense a cero moito máis a miúdo e por iso os números binarios son tan longos: unhas 3,3 veces máis ca os decimais.

UN SÍMBOLO NOVO *como se le e como se escribe?*

Un número en binario **non se le como se lería en decimal**. O `1011` non é «mil once»: é «un, cero, un, un» e vale once.

Cando faga falta distinguir, escríbese a base cun subíndice:

$$1011_{(2)} = 11_{(10)}$$

Lese «un cero un un en base dous é igual a once en base dez». Se non hai subíndice, dáse por suposto que estamos en base dez, coma sempre.

Ás columnas dun número binario chámaselles **bits**, do inglés *binary digit*. O `1011` ten catro bits.

Cada columna vale unha potencia de dous, igual que en decimal cada columna vale unha potencia de dez. E tamén medran de dereita a esquerda, como estamos afeitos:

$$\ldots,\quad 1024,\quad 512,\quad 256,\quad 128,\quad 64,\quad 32,\quad 16,\quad 8,\quad 4,\quad 2,\quad 1$$

EXEMPLO 3.1 *ler `1011` columna a columna*

Faise exactamente o mesmo que co 3 527 do principio do capítulo. Alí as columnas valían 1000, 100, 10 e 1; aquí valen 8, 4, 2 e 1. Escríbense os valores enriba e as cifras debaixo:

a columna vale	8	4	2	1
a cifra é	`1`	`0`	`1`	`1`
achega	8	0	2	1

Súmase o que achega cada columna:

$$1011_{(2)} = 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 8 + 0 + 2 + 1 = 11$$

Pon ao lado a do principio e mira que é a mesma frase:

$$3\,527 = 3 \times 1000 + 5 \times 100 + 2 \times 10 + 7 \times 1$$

Cambian os valores das columnas. A regra non cambia nada.

UN POUCO DE HISTORIA *a base dous xa pasou pola escola*

Se estudaches a Educación Xeral Básica nos anos setenta, é posíbel que isto che soe. O primeiro autor lembra que a numeración en base dous se explicaba en segundo de EXB e que nin o alumnado nin a maioría do profesorado sabían para que servía.

Non era unha ocorrencia local. A partir da Lei xeral de educación de 1970, os programas educativos incorporaron a chamada matemática moderna: conxuntos, relacións e estruturas. O movemento era internacional e bebía da matemática estruturalista do grupo francés Bourbaki. Este tivera o seu momento simbólico en 1959, nun seminario en Royaumont no que Jean Dieudonné lanzou o berro de «abaixo Euclides». Os estudos sobre aquela reforma indican tamén que o profesorado non estaba preparado para ela e que os cursos de adaptación foron pouco axeitados [146]. A reacción posterior e o que se propuxo no seu lugar contouno Miguel de Guzmán [61].

O resultado foi o que cabía esperar: xeracións que aprenderon a escribir números en base dous como un exercicio máis, sen que ninguén lles dixese que estaban aprendendo a linguaxe na que ía funcionar case todo o que usarían de adultas. Este capítulo é, en parte, a explicación que faltou entón.

E aquí atopámonos cun vello coñecido. No capítulo 1, para multiplicar 23 × 45, fomos partindo o 23 pola metade ata chegar ao un e resultou que estabamos descompoñendo o 23 en 16 + 4 + 2 + 1. Pois ben:

$$23 = 16 + 4 + 2 + 1 = 10111_{(2)}$$

Aquela columna de metades **estaba escribindo o número en binario** sen dicilo. Non era unha coincidencia: era a mesma operación.

EXEMPLO 3.2 *pasar o* 45 *a binario*

Dividimos entre dous unha e outra vez, anotando o resto:

número	metade	resto
45	22	**1**
22	11	**0**
11	5	**1**
5	2	**1**
2	1	**0**
1	0	**1**

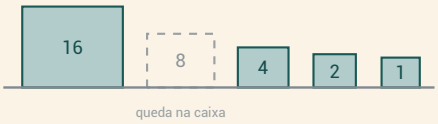
Agora lense os restos **de abaixo a arriba**: 101101. E, en efecto,

$$32 + 8 + 4 + 1 = 45$$

Por que de abaixo a arriba? Porque o primeiro resto que sae é o da columna das unidades, que se escribe á dereita; o último é o da columna maior, que vai á esquerda.

MÍRAO *unha caixa de pesas*

Hai unha imaxe que fai isto obvio. Imaxina unha balanza e unha caixa con pesas de 1, 2, 4, 8 e 16 gramos, unha de cada.



16 + 4 + 2 + 1 = 23
10111

Para pesar 23 gramos pos a de 16, a de 4, a de 2 e a de 1 e deixas a de 8 na caixa. Escribir un 1 ou un 0 en cada columna é exactamente anotar cales pesas usaches.

Con esas cinco pesas podes conseguir calquera peso enteiro de 1 a 31 gramos e cada peso dunha soa maneira. Esa é toda a idea do sistema binario.

POR QUE FUNCIONA? *por que sempre se pode e dunha soa maneira?*

Que se poida sempre vese co procedemento máis simple: colle a pesa máis grande que non se pase, pon esa pesa na balanza e segue co que falta.

Se queres 23: a maior que non se pasa é 16, porque a seguinte pesa sería a de 32 e xa nos pasariamos; quedan 7. A maior que non se pasa de 7 é 4, porque a seguinte sería a de 8 e tamén nos pasariamos; quedan 3. Despois 2; queda 1. Despois 1; queda 0. Rematou. Este método sempre remata porque o que falta diminúe en cada paso e nunca é negativo.

Que sexa *dunha soa maneira* é un pouco máis fino e o argumento é bonito: fíxate en que 1 + 2 + 4 + 8 = 15, que é *menos* ca 16. En xeral, a suma de todas as pesas menores ca unha dada sempre queda xusto un gramo por debaixo dela. Polo tanto, se o peso que buscas chega a 16, **tes** que usar a de 16: sen ela, aínda que uses todas as demais, non chegas. E o mesmo argumento vale despois para cada pesa, unha tras outra. Non hai marxe de elección en ningún paso.

EN PALABRAS SINXELAS *canto cabe nun puñado de bits?*

Cada bit que engades **duplica** o que podes representar, porque engades unha pesa que vale tanto como todas as anteriores xuntas máis unha.

Con 8 bits tes $2^8 = 256$ combinacións distintas: lidas como números, do 0 ao 255. A oito bits chámaselle un **byte** e é a unidade coa que se mide o espazo ou a memoria dentro dun ordenador.

Con 16 bits, 65 536. Con 20 bits xa pasas do millón (1 048 576).

Por certo: aí tes a orixe dunha confusión que igual notaches. 1 024 é 2^{10} e parécese moito a mil, así que durante décadas houbo quen lle chamou «quilo» ao 1 024. Non é o que di a norma, que reserva *quilo* para mil e ten un prefixo á parte, *quibi*, para o 1 024. Pero a confusión quedou instalada e é a razón de que un disco de «500 gigas» apareza en Windows con 465. Contámolo enteiro no capítulo 4.

PRÓBAO TI 3.1 ★☆☆

Pasa a binario o ano en que naciches, polo método das metades e os restos do exemplo de pasar o 45 a binario.

Despois fai a comprobación ao revés: suma as potencias de dous correspondentes aos uns que che saíron e mira se recuperas o ano. Cantos bits fixeron falta?

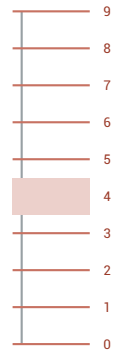
resposta na páx. 369 →

3.3 Por que os aparellos escolleron o dous

Chegados aquí podería parecer que o binario é unha comodidade matemática. Non o é. É unha decisión de enxeñaría e a razón é moi térrea.

Dentro dunha máquina, un número ten que estar gardado *nalgunha cousa física*: a tensión nun cable, a carga nun condensador, a imanación dun anaco de material. E todas esas cousas físicas teñen ruído. A tensión oscila coa temperatura, cos aparellos do lado, coa idade do compoñente.

Supón que queres gardar unha cifra decimal nun cable que vai de 0 a 5 voltios. Necesitas dez niveis distintos.



dez niveis
marxe: 0,28 V



dous niveis
marxe: 2,5 V

Canto máis grande é a franxa de cor, máis ruído aguanta o sinal sen que se confunda a cifra.

POR QUE FUNCIONA? *a conta da marxe de seguridade*

Con dez niveis repartidos entre 0 e 5 voltios, a distancia dun nivel ao seguinte é

$$\frac{5}{9} \approx 0{,}56 \text{ voltios},$$

e como cada nivel ten veciños polos dous lados, a tensión só se pode desviar a metade diso, uns **0**,**28** voltios, antes de que se confunda cunha cifra distinta.

Con dous niveis, en cambio, o corte está no medio: calquera valor por debaixo de 2,5 voltios é un 0 e calquera valor por riba é un 1. A tensión pode desviarse ata **2**,**5** voltios sen que haxa erro.

$$\frac{2{,}5}{0{,}28} \approx 9$$

Case nove veces máis tolerancia ao ruído, coa mesma electrónica. (É unha conta idealizada: nos circuítos reais as cifras cambian, pero a vantaxe mantense.) E iso non é un detalle: unha máquina que fai miles de millóns de operacións por segundo, se se equivocase unha vez de cada mil millóns, equivocaríase varias veces por segundo. A fiabilidade non é un luxo aquí, é a condición para que a cousa funcione.

OLLO! *non é porque as máquinas sexan «simples»*

Óese moitas veces que os ordenadores usan ceros e uns «porque son máquinas moi simples e só entenden si ou non». É unha explicación circular e non di nada.

A razón verdadeira é a contraria: **o binario escolleuse porque é robusto**, non porque sexa pobre. Poderíanse construír máquinas que traballasen en base dez e construíronse algunhas nos anos corenta e cincuenta. Perderon por fiabilidade e por custo, non por falta de imaxinación. E mesmo esas, por dentro, xuntaban pezas de dous estados para representar cada cifra decimal.

E convén non agochar un matiz, porque o dous non gañou en todas partes. Nas memorias flash de hoxe, as das memorias USB e as dos discos modernos, cada cela xa non garda un bit: distingue catro, oito ou dezaseis niveis

de carga. Nas comunicacións pasa o mesmo. Pero mira dúas cousas. O número de niveis segue a ser unha potencia de dous, así que o que sae de alí segue a estar feito de bits. E iso só se pode facer engadindo por riba un sistema que detecta e corrixe os erros que aparecen, é dicir, **pagando á parte a fiabilidade que o binario daba de balde**. Onde se calcula, que é do que vai este libro, todo segue funcionando con dous niveis.

Pagamos un prezo por esa robustez, aínda que menor do que parece. Os números quedan máis longos, pero só por un factor fixo: como moito, un pouco máis de 3,3 veces. E en calquera base hai números que non se poden escribir exactos (a raíz de dous non o é en ningunha); ao pasar a binario só cambia cales o son e cales non. Diso vai o final deste capítulo.

UN POUCO DE HISTORIA *dous séculos e medio de adianto*

En 1703, Leibniz publicou na Académie Royale des Sciences de París unha memoria titulada Explicación da aritmética binaria, que se serve só dos caracteres 0 e 1 [88]. Describía o sistema enteiro: como contar, como sumar, como multiplicar. Non tiña ningunha aplicación práctica á vista, aínda que Leibniz lla buscase. Era, case literalmente, unha curiosidade.

Pasaron douscentos trinta e catro anos ata que servise para algo. En 1937, un estudante de vinte e un anos chamado Claude Shannon presentou no MIT unha tese de mestrado na que amosaba que a álxebra que George Boole inventara en 1854 para tratar a lóxica [20] describía exactamente o comportamento dos circuítos de relés [139]. Verdadeiro e falso por un lado, pasa e non pasa polo outro: a mesma matemática.

Esa tese adoita citarse como a máis influente do século XX e non é esaxeración. Sen ela non habería ordenadores tal e como os coñecemos. E o material de partida levaba douscentos anos gardado nun caixón, sen servir para nada.

Paga a pena non pasar por riba disto, porque é un argumento que fai falta cada vez que alguén pregunta para que serve unha investigación. Leibniz non estaba a preparar o terreo para nada: estaba a mirar un sistema de numeración porque lle parecía fermoso. Se en 1703 se lle pide que xustifique a utilidade daquilo, non tería qué dicir. E sen aquilo non habería nada do que conta este libro.

Abraham Flexner escribiu en 1939 un ensaio enteiro sobre isto, titulado precisamente A utilidade do coñecemento inútil [46]. A súa tese é a que acabamos de ver en miniatura: ***o coñecemento non ten que xustificar por adiantado para que vai servir****, entre outras cousas porque quen o produce non o pode saber. O único saber que de seguro non serve para nada é o que non se ten.*

UN POUCO DE HISTORIA *dous signos, moito antes de Leibniz*

Leibniz tampouco foi o primeiro en contar con dous signos. Na India, un tratado de métrica atribuído a Pingala (de data incerta, pero xa citado no século IV da nosa era) enumeraba os ritmos da poesía sánscrita combinando sílabas breves e longas, dun xeito moi parecido a contar en binario [159]. E cando Leibniz recibiu desde China, a través do xesuíta Joachim Bouvet, un diagrama dos hexagramas do I Ching (figuras de seis liñas, enteiras ou partidas), viu nel a súa aritmética e engadiulle ao título da memoria que o binario daba «o sentido das antigas figuras chinesas de Fohy». Os hexagramas non se pensaran como números, pero a coincidencia de forma é real.

3.4 Sumar en binario: unha táboa de catro liñas

Aquí vén a recompensa de todo o anterior e é máis grande do que parece.

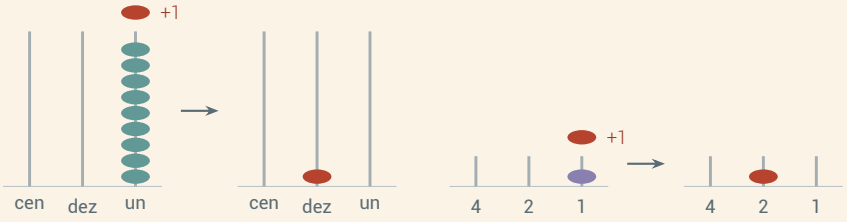
Para sumar en decimal hai que saber unha táboa de **cen** entradas: desde 0 + 0 e 0 + 1 ata 9 + 9. Calquera de nós a memorizou na infancia e custounos meses. En binario, a táboa enteira é esta:

			escribes	levas
0	+	0	0	0
0	+	1	1	0
1	+	0	1	0
1	+	1	0	1

Catro liñas. Só hai que reter unha cousa: **un máis un é cero e levas unha**, que é o mesmo que pasa en decimal con 9 + 1, só que aquí chega moito antes.

MÍRAO *qué quere dicir «levar»? Dous ábacos*

Cun ábaco diante, *levar* significa exactamente o que di: coller unha conta e levala á variña da esquerda.



base dez: 9 + 1 = 10 **base dous:** 1 + 1 = 10

No ábaco de dez cada variña admite ata nove contas. Se nas unidades xa hai nove e chega outra, non cabe: retíranse as dez e ponse unha soa conta na variña das decenas. Esa conta que pasa dunha variña á seguinte é a que *levas*.

No ábaco de dous cada variña só admite unha conta (por iso son tan curtas). A levada chega en canto se xuntan dúas: 1+1 xa non cabe, retíranse as dúas e lévase unha á variña seguinte. É o mesmo movemento, moito antes.

Paga a pena deterse na palabra, porque é das que máis problemas lles deu ás xeracións de nenos e nenas. Para unha criatura de seis anos, levar é levar algo a algures. Co ábaco diante, a palabra describía o que facían as mans. Cando o ábaco desapareceu das aulas e quedou a palabra, «levo unha» pasou a ser unha fórmula que se recitaba sen ver qué levaba nin a onde.

Noutras partes do mundo o ábaco non desapareceu. En Xapón o *soroban*, parente do *suanpan* chinés, seguiu sendo unha ferramenta de uso diario. En novembro de 1945, nun desafío celebrado en Xapón, un campión de *soroban* gañoulle catro de cinco roldas ao operador de calculadora eléctrica con máis sona das tropas estadounidenses alí destinadas [70].

EXEMPLO 3.3 *sumar* 1011 *e* 0110

Son o 11 e o 6. Sumamos columna a columna, de dereita a esquerda, como sempre.

1	1	1			levadas
	1	0	1	1	11
+	0	1	1	0	6
1	0	0	0	1	17

Columna das unidades: 1 + 0 = 1, non levamos nada. Columna dos dous: 1 + 1 = 0 e levo 1. Columna dos catros: 0 + 1 máis a levada 1, outra vez 0 e levo 1. Columna dos oitos: 1 + 0 máis a levada, 0 e levo 1. E esa última levada escríbese diante.

Resultado: $10001_{(2)}$, que é 16 + 1 = 17. E, en efecto, 11 + 6 = 17.

Coa multiplicación pasa algo aínda mellor. A táboa de multiplicar binaria tamén ten catro liñas e as catro son triviais:

$$0 \times 0 = 0, \qquad 0 \times 1 = 0, \qquad 1 \times 0 = 0, \qquad 1 \times 1 = 1$$

POR QUE FUNCIONA? *por que multiplicar é correr as cifras?*

Ao multiplicar como na escola, cada cifra do multiplicador dá unha fila da conta (o que se chamaba *produto parcial*) e ao final súmanse todas. En binario, cada cifra do multiplicador só pode ser 0 ou 1. Se é 0, a súa fila é toda ceros. Se é 1, a súa fila é *o outro número tal cal*, desprazado tantas posicións como indique a columna.

E desprazar unha posición á esquerda é multiplicar por dous, exactamente igual que en decimal desprazar unha posición é multiplicar por dez. (1011 vale 11; 10110 vale 22.)

Así que multiplicar en binario redúcese a: **desprazar e sumar**. Sen táboa, sen memorizar nada.

Decatácheste? Iso é, palabra por palabra, o método exipcio das metades e os dobres do capítulo 1. Non era un truco antigo: era o algoritmo que ía acabar dentro de todos os procesadores do mundo, tres mil cincocentos anos antes de que houbese procesadores.

PRÓBAO TI 3.2 ★★☆

Suma en binario 1101 e 1011 sen pasalos a decimal, columna a columna, anotando as levadas. Despois comproba o resultado convertendo os tres números a decimal.

Pista: vas ter que levar catro veces seguidas. É un bo exemplo de que os números binarios son longos pero as contas son sinxelas.

resposta na páx. 370 →

3.5 E os números negativos?

Queda un oco evidente: ata aquí todos os números foron positivos. Dentro da máquina non hai signo menos, do mesmo xeito que non hai coma: hai bits.

A primeira idea é a que terías ti se só tiveses ceros e uns: **que o primeiro bit diga o signo** (un 0 para os positivos, un 1 para os negativos). Funciona, pero ten dous inconvenientes: con oito bits só se chega do −127 ao 127 e o cero ten dúas escrituras, +0 e −0, o que complica os circuítos. Coa maneira de facelo que

imos ver agora, en vez de chegar do 0 ao 255, chégase do −128 ao 127: a mesma cantidade de números, corrida para abaixo e cun só cero. (Os números con coma, en cambio, conservan o +0 e o −0 a propósito, porque serve para saber por que lado se chegou ao cero.)

MÍRAO *por que restar é sumar?*

O detalle bonito é como se representa o negativo. Non se garda «o mesmo número cunha marca»: gárdase o que lle falta para dar a volta ao contador.

Pensa nun contador de quilómetros de tres cifras, dos de antes. Se marcas 000 e andas para atrás un quilómetro, queda 999. Así que, nese contador, 999 *é* o −1: se lle sumas 1 volves ao 000.

En binario faise exactamente igual e ten unha consecuencia enorme para a máquina: **non fai falta un circuíto de restar**. Restar 5 é sumar o representante do −5 e deixar que o contador dea a volta. Un dos capítulos seguintes constrúe un sumador con interruptores; grazas a este truco, ese mesmo sumador xa sabe restar.

Esta maneira de facelo chámase **complemento a dous** e é a que usan para os enteiros practicamente todas as máquinas desde hai sesenta anos.

3.6 E os números con coma?

Ata aquí, enteiros. Os números con decimais tamén se gardan en binario, pero cun truco que ten un prezo: **moitos non se poden gardar exactos**. Xa o adiantamos no capítulo 1: 0,1, que en decimal é limpísimo, en binario é periódico e a máquina ten que cortalo. O que garda non é 0,1: é algo extraordinariamente parecido.

Hai unha regra que di cales se poden gardar exactos. Unha fracción, unha vez simplificada, escríbese exacta nunha base só se o seu denominador non ten máis factores primos cós da base. En decimal, 10 = 2×5; en binario, só o 2. Por iso $1/10$ é exacta en decimal e periódica en binario; $1/3$, en cambio, é periódica nas dúas. (Os babilonios, coa base 60 = 2 × 2 × 3 × 5, escribían exactos os terzos.) E os números que non son fraccións, coma a raíz de dous, non se poden escribir exactos en ningunha base.

PARA QUE? *números con coma nun libro sobre intelixencia artificial*

Para que o ordenador poida representar medidas e non só contas: distancias, tempos, probabilidades ou os pesos dunha rede neuronal, que teñen parte fraccionaria e se moven en escalas moi distintas. Os enteiros só serven para contar unidades.

Dentro dos modelos de linguaxe máis grandes hai centos de milleiros de millóns de números e **case todos son deste tipo**: aproximacións gardadas cun número finito de bits. A lección do capítulo 1 segue válida en todos os andares do edificio: *a máquina aproxima e ningunha parte dela cho di*. Para o que adoitamos facer, esa aproximación abonda; o problema aparece cando alguén dá por exacto o que non o é.

UN CHANZO MÁIS *a coma flotante*

O truco chámase **coma flotante**. O número gárdase coma na notación científica: unhas poucas cifras significativas e, á parte, onde vai a coma (por exemplo, $6{,}02 \times 10^{23}$). Así caben no mesmo espazo números enormes e números diminutos, a cambio de cortar as cifras que non collen.

Por que 0,1 non se pode gardar exacto, cómo se escribe un número con coma en binario e cómo se fai a conta paso a paso relátase con calma no apéndice B. O resto do libro non o necesita.

PÉNSAO UN MOMENTO *se o binario é tan bo, por que non o usamos nós?*

Se o binario ten tantas vantaxes, por que seguimos contando en base dez? A resposta está máis en nós ca nos números. A memoria de traballo humana manexa ben uns poucos elementos á vez, arredor de sete [98]: os días da semana, por exemplo. En binario abondarían dúas cifras, pero os números serían unhas 3,3 veces máis longos. Un número de teléfono de nove cifras pasaría a ter trinta. E mesmo con nove agrupámolas para lembralas, de tres en tres ou un grupo de tres e despois tres de dous: agrupar fai máis doado ler, lembrar e comparar.

Cando se programa faise o mesmo. Non se len os bits un a un: agrúpanse en bloques de catro e escríbese cada bloque cunha soa cifra do sistema *hexadecimal*, que vai do 0 ao 9 e despois segue con A, B, C, D, E e F. Así, `1111101000` (o 1 000 en decimal) agrúpase como `11 1110 1000` e pasa a ser `3E8`, moito máis doado de ler, lembrar e comparar.

O ESENCIAL DESTE CAPÍTULO

- O **valor posicional** é o invento; a base dez é, probabelmente, herdanza dos dez dedos, non unha necesidade matemática.
- En base dous cada columna vale unha potencia de dous e todo número se descompón nelas **dunha soa maneira**: é a caixa de pesas.
- Esa descomposición xa apareceu no capítulo 1: a columna das metades estaba escribindo o número en binario.
- Escolleuse o dous **por fiabilidade**: con dous niveis a marxe de ruído é nove veces maior ca con dez, coa mesma electrónica.
- A táboa de sumar binaria ten **catro liñas** en vez de cen e multiplicar redúcese a desprazar e sumar.
- En binario, **multiplicar por dous é correr as cifras un lugar á esquerda e dividir por dous é correlas á dereita**. Máis sinxelo non pode ser: é a operación máis barata da máquina.
- Os números con decimais gárdanse moitas veces **aproximados**: dentro da máquina, 0,1 non se pode gardar exacto (detalles no apéndice B).

E AGORA?

Xa sabemos que dentro só hai números e como se escriben. Pero as palabras non son números, nin as fotos, nin a música. Alguén tivo que decidir en que número se converte a letra «ñ» e esa decisión aínda nos afecta.

Capítulo 04

Letras, cores e cancións

PARA ARRANCAR

Se dentro só hai números, en que número se converte a letra «ñ»? E unha fotografía da túa avoa? E a canción que estás a escoitar agora mesmo? Alguén tivo que decidilo e as decisións que tomou aínda nos afectan.

4.1 Un número para cada letra

Xa sabemos gardar números. Pero unha letra non é un número, unha foto tampouco e un acorde menos aínda. Para que entren na máquina hai que *converter todo en números* e converter quere dicir aquí unha cousa moi concreta: **poñerse de acordo**.

Non hai ningunha propiedade da letra A que a faga corresponder a un número determinado. A correspondencia hai que inventala, escribila nunha táboa e conseguir que todo o mundo use a mesma. É un acordo internacional, non un descubrimento.

EN PALABRAS SINXELAS *o teclado explícao mellor ca ninguén*

Mira o teclado que tes diante. Cada letra ten asignada unha tecla, é dicir, unha posición. Non hai ningunha razón matemática para que a `A` estea á esquerda, a `P` á dereita ou a `M` na fila de abaixo: poderían estar noutro sitio e o teclado funcionaría igual de ben. E non é unha suposición: hai teclados con outras distribucións e funcionan perfectamente. O que usamos aquí chámase QWERTY, polas súas seis primeiras letras; en Francia úsase o AZERTY e en Alemaña o QWERTZ, que cambian de sitio unhas cantas teclas. E hai quen escribe nun DVORAK, deseñado nos anos trinta para que as letras máis frecuentes queden baixo os dedos que menos se moven. O que fai falta non é unha distribución concreta: é que quen escribe e quen le use a mesma. Cando premes unha tecla, ti sabes que letra querías e a máquina sabe cal ten que interpretar.

Co ASCII pasa exactamente o mesmo, só que en vez de asignarlles ás letras posicións nun teclado, asígnalles números.

O primeiro tratado importante foi o **ASCII**, de 1963. Reserva sete bits para cada carácter, o que dá 2^7 = 128 números distintos. Uns poucos deles:

carácter	número	carácter	número	carácter	número
espazo	32	`A`	65	`a`	97
`0`	48	`Z`	90	`z`	122
`9`	57				

EXEMPLO 4.1 *escribir «Ola» en números*

A palabra `Ola` ten tres caracteres, así que son tres números:

`O` → 79, `l` → 108, `a` → 97

E, en binario, tres bytes:

```
01001111 01101100 01100001
```

Iso é literalmente o que hai gardado nun ficheiro de texto que conteña a palabra `Ola`: vinte e catro bits. Nin letras, nin saúdo, nin galego. Tres números.

Agora fíxate nunha cousa que parece casual e non o é.

POR QUE FUNCIONA? *por que a maiúscula e a minúscula se levan exactamente 32?*

Mira outra vez a táboa: `A` é 65 e `a` é 97. A diferenza é 32. E $32 = 2^5$, é dicir, unha potencia de dous, que é como dicir **unha soa columna binaria**.

```
A   01000001
a   01100001
```

Son idénticos agás nun bit. Isto non é casualidade: foi deseñado así. Quen fixo a táboa colocou as letras de modo que **cambiar de maiúscula a minúscula fose apagar ou acender un interruptor**, unha das operacións máis baratas que existen. O mesmo truco está nos díxitos: o carácter `7` vale 55 e restándolle o `0`, que vale 48, sae o número 7.

É unha táboa arbitraria, si, pero está *ben pensada*. As correspondencias non se escollen ao chou: escóllense para que as operacións que vas facer moito saian baratas.

UN POUCO DE HISTORIA *por que a letra A non é o número 1?*

Podería parecer máis natural empezar por `A` = 1. Non se fixo e o motivo é que a táboa ASCII non se deseñou só para procesar texto, senón tamén para transmitilo por teletipo.

Os primeiros trinta e dous símbolos non son realmente caracteres que se vexan: son ordes para a máquina receptora. Salto de liña, retorno de carro, campá (o número 7 facía soar un timbre de verdade), fin de transmisión. Son restos dun mundo de cintas perforadas e máquinas que escribían soas.

Aínda os arrastramos. Cando abres un ficheiro feito noutro sistema operativo e as liñas saen pegadas ou con cadradiños ao final, estás vendo unha discusión de 1963 sobre se o final de liña é un único símbolo ou son dous. Se o teu ordenador considera que é un só e o ficheiro trae dous, o que sobra aparece como ese cadradiño.

*O que si se respectou foi a **orde alfabética**. E iso vén de moito máis atrás ca 1963: vén de que alguén, nalgún lugar do Mediterráneo oriental hai uns tres mil anos, puxo os símbolos nunha táboa nunha certa orde e esa orde*

sobreviviu, con cambios, a todas as culturas polas que pasou a escritura. Un alfabeto é, basicamente, unha táboa de caracteres que alguén ordenou. O ASCII limitouse a estender o alfabeto que xa tiñamos para incluír os símbolos do teletipo e algúns outros. Hoxe a táboa de referencia é ***Unicode****, construída como unha extensión do ASCII para representar os caracteres e os símbolos de todo o mundo (volvemos a ela na sección seguinte).*

E non é por respecto. É por comodidade: se as letras van numeradas na orde do abecedario, ***comparar dúas palabras é comparar dous números*** *e ordenar unha listaxe faise co mesmo procedemento que serve para ordenar prezos. Ninguén ten que escribir un algoritmo especial para as palabras.*

Iso si, funciona ata que aparecen os acentos. Se se comparan os números tal cal, o `Á` *ten un número moi por riba do* `Z` *e verás un Álvarez detrás dun Zapatero. Para ordenar de verdade como manda o dicionario fai falta un conxunto de regras á parte, distinto para cada lingua, porque en galego o á vai co a e noutros idiomas non. Unicode ten decenas de aes distintos e ningún deles sabe onde lle toca ir.*

4.2 O galego dentro da máquina

O ASCII orixinal, de sete bits, ten cento vinte e oito símbolos. Chegan?

Para o inglés, xustos. Para case calquera outra lingua, non. Non hai `ñ`, nin `á`, nin `ç`, nin `ü`. E iso é só o alfabeto latino: non hai grego, nin cirílico, nin árabe, nin as decenas de miles de caracteres chineses.

A saída durante décadas foi un remendo. Como os bytes teñen oito bits e o ASCII só usaba sete, quedaban outros 128 números libres e en cada país enchéronse co que se necesitaba. O resultado foi un caos perfectamente previsíbel: **o mesmo byte significaba unha letra distinta segundo a táboa que usase quen abría o ficheiro**. Un texto galego aberto cunha táboa rusa daba ruído.

En 1991 apareceu Unicode, cunha idea sinxela e ambiciosa: *un número único para cada carácter de cada sistema de escritura do mundo*. Hoxe define máis de cen mil caracteres. O noso `ñ` é o 241 e o `á` é o 225.

UN SÍMBOLO NOVO *carácter e byte non son o mesmo*

Aquí hai unha distinción que convén non perder, porque vai reaparecer no capítulo 8 e alí será importante.

Un **carácter** é unha letra, un signo, unha ideografía. Un **byte** son oito bits. Durante a época do ASCII coincidían e collemos o vicio de falar deles como se fosen o mesmo. Xa non coinciden.

Na codificación que usa hoxe case todo o mundo, chamada **UTF-8**, cada carácter ocupa *entre un e catro bytes*: as letras do ASCII seguen ocupando un (para que os ficheiros vellos sigan valendo), as letras acentuadas ocupan dous, o símbolo do euro tres e algúns signos catro.

Unha precisión: Unicode é o catálogo de caracteres, cada un co seu número. UTF-8 é unha *codificación*: di como se converte cada un deses números en bytes. O ASCII era as dúas cousas á vez, catálogo e codificación, só para 128 caracteres. UTF-8 estendeuno e segue sendo compatible con el: eses 128 caracteres ocupan nel un só byte e co mesmo valor.

EXEMPLO 4.2 *canto ocupa a palabra «mañá»?*

Catro caracteres. Pero:

carácter	bytes	valores
`m`	1	109
`a`	1	97
`ñ`	2	195, 177
`á`	2	195, 161
total	**6**	

Catro letras, seis bytes. Un cincuenta por cento máis de espazo cá mesma palabra escrita sen til nin acento, *mana*.

OLLO! *de onde saen os «Ã±»?*

Seguro que viches algunha vez un texto onde debía poñer *mañá* pon `maÃ±Ã¡`. Agora podes explicar exactamente por que.

A `ñ` gardouse en UTF-8 como dous bytes: 195 e 177. Pero o programa que abriu o ficheiro non tiña información de que era UTF-8 e tratouno coma unha daquelas táboas vellas de oito bits. Nesa táboa, o byte 195 é `Ã` e o 177 é `±`. Así que, obedientemente, imprimiu dous caracteres onde había un.

Non hai ningunha corrupción de datos: os bytes están intactos. O que falla é que **a codificación empregada (UTF-8 ou unha das táboas vellas) non está nestes bytes**. Adoita indicarse noutros bytes, como na cabeceira das páxinas web, pero se as dúas partes non usan a mesma codificación, sae isto.

Podes velo ti: preme co botón dereito do rato sobre calquera páxina web e escolle «Ver código fonte da páxina». Preto do principio atoparás unha liña coma `<meta charset="UTF-8">`.

PARA QUE? *isto non é un detalle técnico neutro*

Durante trinta anos, escribir na túa lingua nun ordenador era doado se escribías en inglés, incómodo se escribías en galego e case imposíbel se escribías en tamil. As decisións de codificación decidiron as linguas que circulaban ben pola rede e as que non.

Unicode arranxou a maior parte diso, pero non todo. O UTF-8 segue premiando o inglés e convén dicir canto exactamente, sen esaxerar: as letras inglesas ocupan *sempre* un byte e as nosas tamén, agás as que levan til ou diérese e mais o eñe, que ocupan dous. Nun texto en galego iso é unha minoría das letras, así que en espazo a diferenza é pequena. Onde deixa de ser pequena é máis adiante: no capítulo 8 veremos que na intelixencia artificial xenerativa a mesma frase custa máis en galego por outra razón. E alí a diferenza xa non é pequena.

UN POUCO DE HISTORIA *a primeira persoa que viu isto e viuno en 1843*

Todo este capítulo defende unha idea: que cousas que non son números (letras, cores, sons) se poden converter en números e meter nunha máquina. A idea ten dono e ten data.

En 1842, o enxeñeiro italiano Luigi Menabrea publicou en francés unha descrición da máquina analítica que Charles Babbage proxectara e nunca chegou a construír. Traduciuna ao inglés a matemática ***Ada Lovelace****, que levaba anos estudando a máquina e entendíaa mellor ca ningunha outra persoa fóra do propio Babbage. Á tradución engadiulle sete notas propias, que xuntas ocupan máis có texto orixinal. Asinounas só coas iniciais* A.A.L. *[96]. As iniciais non eran modestia: unha muller asinando sobre matemáticas en 1843 era outra cousa.*

Na nota G está o que adoita citarse: un procedemento completo, paso a paso, para que a máquina calculase unha certa serie de números. É o primeiro programa publicado para unha máquina que aínda non existía.

Pero o que nos interesa aquí está na nota A e é máis fondo. Babbage falaba dunha máquina de facer contas. Lovelace decatouse de que, se as relacións entre as notas musicais se puidesen expresar en números, a mesma máquina podería compoñer música. Non estaba a falar de música por dicir algo: estaba a dicir que unha máquina de números deixa de ser unha máquina de números en canto alguén atopa como interpretar os números como outra cousa. Que é, literalmente, o que fixemos nas páxinas anteriores coas letras e o que imos facer agora coas imaxes e o son.

Tardou máis dun século en ser certo. Lovelace morreu en 1852, aos trinta e seis anos. A máquina analítica non se construíu nunca.

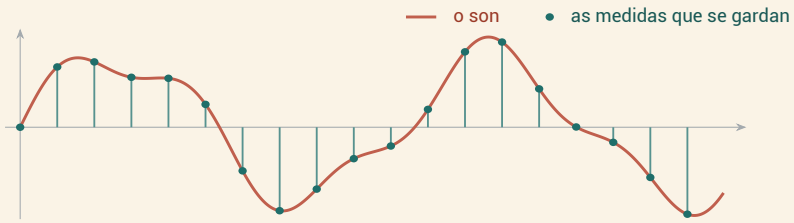
E a mesma persoa deixou escrita a advertencia que abre o capítulo 1 deste libro: «a máquina analítica non ten ningunha pretensión de orixinar nada; pode facer calquera cousa que saibamos ordenarlle que faga». Cento oitenta anos despois seguimos discutindo esa frase.

4.3 O son é unha lista de alturas

O son é presión do aire que sobe e baixa moi rápido. Un micrófono convérteo nunha tensión eléctrica que tamén sobe e baixa: unha curva continua.

Para meter esa curva na máquina faise o único que se pode facer: **medila moitas veces por segundo e anotar os valores**. A iso chámaselle **mostraxe** e o resultado é unha lista de números.

MÍRAO *a escaleira que finxe ser unha curva*



Cada punto é un número. Entre punto e punto non se garda nada: a máquina non sabe o que pasou aí. Reconstrúeo despois supoñendo que a curva era razoábel.

Canto de a miúdo hai que medir? Se mides poucas veces, perdes os sons agudos; se mides moitas, ocupa moito. O criterio é este: **hai que medir máis do dobre de veces por segundo que a frecuencia máis alta que queres conservar**. Se se cumpre, a curva pódese reconstruír enteira a partir das medidas. É o **teorema de mostraxe**, que se asocia aos nomes de Harry Nyquist e Claude Shannon [108, 141].

O oído humano chega, no mellor dos casos, aos 20 000 ciclos por segundo. O dobre son 40 000. Por iso o estándar de audio dixital mide 44 100 veces por segundo: os 40 000 necesarios máis unha marxe para os filtros. O número exacto herdouse dos primeiros gravadores dixitais, que gardaban o son en cintas de vídeo, entre outras razóns técnicas da época. Hoxe tamén se usan 48 000 medidas por segundo (no vídeo e no Blu-ray) e múltiplos coma 96 000 ou 192 000. Por riba dos 40 000, para o oído humano vén dando igual: o importante é poñerse de acordo.

Cada medida gárdase con 16 bits, é dicir, 65 536 alturas posíbeis. Con dúas canles

para o estéreo, a conta por segundo é

$$44\,100 \text{ medidas} \times 2 \text{ bytes} \times 2 \text{ canles} = 176\,400 \text{ bytes},$$

e unha canción de tres minutos ocupa

$$176\,400 \text{ bytes por segundo} \times 180 \text{ segundos} = 31\,752\,000 \text{ bytes} \approx 30{,}3 \text{ MiB}$$

E, con todo, no teléfono unha canción así ocupa uns catro MiB, unhas sete veces menos. Falta algo.

UN SÍMBOLO NOVO *MB e MiB, ou por que o disco sempre parece menor*

Hai dúas maneiras de contar o mesmo e conviven.

Un **megabyte** (MB) son un millón de bytes exactos, seguindo o sistema métrico de sempre. Un **mebibyte** (MiB), coa sílaba *bi* de *binario*, son exactamente $2^{20} = 1\,048\,576$ bytes, unha potencia de dous, o idioma natural da máquina.

A diferenza é dun 4,9 % e vai medrando: un *gibibyte* (GiB) é un 7,4 % maior ca un xigabyte. E aí está a razón do disco que mingua. Un disco vendido como de 500 GB ten 500 000 millóns de bytes, que é o que promete a etiqueta. Divídeos entre 2^{30} e dan 465,7 GiB, que é como os conta o ordenador. Por iso na pantalla pon 465 e non 500: ninguén mente, están a contar coa mesma cifra en unidades distintas.

As dúas escalas non naceron á vez. Primeiro houbo unha soa e era chapuceira: como 1 024 se parece a 1 000, os profesionais da informática colleron prestados os prefixos do sistema métrico e empezaron a chamarlle «quilo» ao 1 024. Funcionou mentres os números eran pequenos. Cando pasaron a ser grandes, o desaxuste deixou de ser un redondeo. Por iso a Comisión Electrotécnica Internacional (IEC) creou en **1998** unha familia de prefixos só para as potencias de dous: *quibi*, *mebi*, *gibi*, coas abreviaturas Ki, Mi, Gi. A norma foi adoptada despois polo IEEE, pola Unión Europea e polo NIST.

Aínda así, non a usa todo o mundo. Os fabricantes de discos contan en potencias de dez, que é o que manda o sistema métrico. macOS e a maioría das distribucións de Linux tamén. Windows segue amosando «GB» onde en realidade mide GiB.

Convén non despachar isto como teimosía. Un sistema operativo que leva décadas funcionando ten esa cifra escrita en milleiros de sitios, dentro e

fóra: en programas doutros, en documentación, en contratos, na cabeza da xente. Cambiarlle o significado a un número que xa está en todas partes rompe compatibilidades por todos os lados e a compatibilidade cara atrás é precisamente unha das cousas que ese sistema vende. Non é que non saiban distinguir: é que arranxalo custa máis do que soluciona.

Neste libro usamos MiB porque así a unidade queda inequívoca:

$$31\,752\,000/1\,048\,576 \approx 30{,}3$$

4.4 Unha imaxe é unha táboa de números

Coas imaxes faise algo semellante ao son, só que as mostras non se toman ao longo do tempo, senón do espazo: percórrese a imaxe de esquerda a dereita e de arriba abaixo e tómanse tantas mostras como se considere. Cada mostra é un cadradiño (un **píxel**) e dise de que cor é.

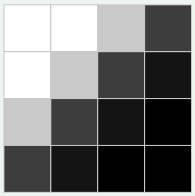
Se a imaxe é en branco e negro, chega un número por píxel: 0 é negro, 255 é branco e no medio os grises.

EXEMPLO 4.3 *unha imaxe de dezaseis píxeis*

Á esquerda, os números. Á dereita, o que se ve.

255	255	200	60
255	200	60	20
200	60	20	0
60	20	0	0

os números



o que se ve

Dezaseis números, dezaseis cadradiños. Nada máis. Se a imaxe fose en cor farían falta tres números por píxel (canto vermello, canto verde e canto azul), é dicir, corenta e oito en total.

Con tres números de 0 a 255 pódense representar

$$256 \times 256 \times 256 = 16\,777\,216$$

cores distintas: máis de dezaseis millóns, que son tres bytes por píxel, ou 24 bits. É máis do que, na práctica, somos quen de distinguir e por iso serve.

Tamén aquí houbo que poñerse de acordo e eses 24 bits son o estándar dos aparellos de consumo: pantallas, ficheiros, impresoras. Onde importa distinguir máis, úsase máis. As radiografías gárdanse con 12 bits ou máis por píxel, máis de catro mil tons de gris. E hai sensores de cámara que miden 16 bits en cada canle.

MÍRAO *achégate ata que se rompa*

Colle unha foto no teléfono e amplíaa ata o final, ata que xa non se vexa a escena senón os cadradiños. Iso que ves *é* a foto: sempre foi iso.

O que pasa é que a distancia normal de visión e o tamaño da pantalla fan que o ollo mesture os cadradiños e vexa unha cara. A imaxe continua non está en ningures: está na túa cabeza, non no ficheiro.

Botemos a conta do que ocupa unha foto de teléfono. Unha de doce megapíxeis son 4 000 × 3 000 = 12 millóns de píxeis. A tres bytes cada un:

$$12\,000\,000 \text{ píxeis} \times 3 \text{ bytes} = 36\,000\,000 \text{ bytes},$$

uns 34,3 MiB. E outra vez o mesmo desaxuste: cando miras o ficheiro no teléfono ocupa uns catro MiB, oito ou nove veces menos.

UN POUCO DE HISTORIA *o cine e a televisión xa facían mostraxe*

O cine fai unha mostraxe no tempo: 24 *fotogramas por segundo, cada un unha foto quieta. A televisión analóxica facía ademais unha mostraxe no espazo, pero só en vertical: no sistema PAL, o que se usou en boa parte de Europa desde finais dos anos sesenta, cada imaxe percorríase en* 625 *liñas. Ao longo de cada liña o sinal seguía sendo continuo; cando se dixitalizou, cada liña pasou a gardarse en* 720 *puntos.*

E hai unha curiosidade. O cine roda a 24 *imaxes por segundo, pero a televisión PAL emitía* 25 *(noutros países, arredor de* 30*). Nos países co PAL, a solución máis simple foi pasar as películas un pouco máis rápido: un* 4 % *máis curtas e coa voz e a música un chisco máis agudas ca no cine.*

4.5 Comprimir é tirar o que non se nota

A diferenza entre os trinta e tantos MiB e os catro, tanto na canción coma na foto, chámase **compresión** e hai dúas clases moi distintas.

A primeira non perde nada. Se nunha imaxe hai douscentos píxeis brancos seguidos, en vez de gardar douscentos números iguais gárdase «branco, douscentas veces». Ao descomprimila recupérase exactamente a orixinal. É a idea dos ficheiros comprimidos de texto: reversíbeis ao cento por cento.

A segunda **tira información a propósito** e é a que fai o traballo de verdade nas fotos e na música. O razoamento é: se hai detalles que o ollo ou o oído non van percibir, para que gardalos?

✔ Sen perdas	⚠ Con perdas
Recupérase o orixinal exacto. Reduce pouco: dúas ou tres veces. Texto, programas, datos.	O orixinal xa non volve. Reduce moito: oito ou dez veces. Fotos, música, vídeo.

As cifras de antes cadran agora. A foto pasa de 36 millóns de bytes a uns catro MiB: unhas 8,6 veces menos. A canción pasa de 31,75 millóns a 4,32 millóns cun MP3 de calidade normal: unhas 7,3 veces menos. En ambos os casos, o que desapareceu **non volve**.

⚠ **OLLO!** *quen decide o que «non se nota»?*

Esta é a pregunta interesante e non ten unha resposta técnica.

O que se tira non se escolleu ao chou: escolleuse a partir de modelos de como percibimos, construídos con experimentos en persoas. Se hai un son forte, os sons febles inmediatamente posteriores non se oen, así que se tiran. O ollo distingue peor as variacións de cor cás de brillo, así que a cor se garda con menos detalle.

Pero eses modelos describen unha *persoa media* e as persoas medias non existen. Hai quen oe o que os modelos din que non se oe. E as decisións tomáronse en comités hai décadas e hoxe afectan a todo o audio e a todas as imaxes do mundo.

Segunda consecuencia, máis práctica: se comprimes unha imaxe xa comprimida, o dano acumúlase. Cada vez que unha foto pasa por unha aplicación que a volve comprimir, perde un pouco máis. Por iso as imaxes que levan anos circulando por mensaxería teñen ese aspecto lavado.

UN CHANZO MÁIS *como chegou a cor á tele sen deixar fóra as teles en branco e negro?*

Durante moitos anos, antes da televisión dixital, nas casas convivían teles en cor e teles en branco e negro. O sinal que chegaba pola antena era o mesmo para todas, pero cada unha amosaba unha cousa distinta. Como pode ser? A clave foi non tocar o que xa funcionaba e engadir a cor nunha zona do sinal á que as teles vellas case non respondían. Non porque non prestasen atención, senón porque a súa tecnoloxía non o permitía (todo o mundo sabe que as teles non prestan atención: róubana).

1. **A tele en branco e negro só recibía o brillo.** O sinal dicía, liña a liña, canto de claro ou de escuro tiña cada punto da imaxe. Iso chámase *luminancia*.
2. **Na tele en cor ese sinal de brillo quedou igual.** As teles antigas seguían recibindo exactamente o que precisaban para debuxar a imaxe en grises.
3. **A cor viaxaba nunha frecuencia moito máis alta.** A información de cor (a *crominancia*: que tonalidade e canta intensidade) ía montada nunha onda auxiliar situada no extremo alto do sinal. As teles en branco e negro apenas respondían a esas frecuencias e o pouco que lles chegaba convertíase nun patrón de puntiños tan fino que case non se vía.
4. **A tele en cor xuntaba as dúas cousas.** Combinaba o brillo coa cor para reconstruír o vermello, o verde e o azul de cada punto. Un pequeno sinal de referencia en cada liña servíalle para sincronizarse e acertar coas cores.
5. **O truco de fondo era aproveitar como vemos.** O ollo percibe moito mellor os detalles de brillo cós de cor, como vimos na compresión. Por iso a cor podía enviarse con menos detalle sen que se notase e cabía no mesmo canal.

A idea de gastar a precisión só onde o ollo a nota segue moi viva. Nas gafas de realidade virtual úsase o *renderizado foveado*: unhas cámaras seguen o movemento dos ollos e o sistema debuxa con máxima resolución só a zona á que estamos mirando, a que capta a *fóvea*, a parte da retina con máis agudeza visual. O resto da imaxe debúxase con menos detalle.

O sorprendente é o pequena que é esa zona: arredor de dous graos do campo visual, máis ou menos o tamaño da unlla do polgar co brazo estendido. Só aí precisamos verdadeira precisión; fóra dela, o cerebro enche o resto e non notamos a diferenza. Como na tele en cor hai décadas, a tecnoloxía

non intenta ser perfecta en todo: apóiase nos nosos límites.

PRÓBAO TI 4.1 ★☆☆

Fai a conta ti cun vídeo, que é o caso máis espectacular.

Un vídeo de alta definición son 1920 × 1080 píxeis, tres bytes por píxel e 25 imaxes por segundo. Calcula cantos bytes ocuparía *un minuto* sen comprimir.

Despois compárao co que ocupa realmente un minuto de vídeo no teu teléfono. Cantas veces menos é? Que che di iso sobre a cantidade de información que se está tirando?

resposta na páx. 371 →

4.6 O que se perde ao pasar de analóxico a dixital

Recapitulemos, porque debaixo dos dous casos hai a mesma estrutura.

Unha imaxe e unha canción son cousas continuas, ambiguas e inesgotábeis. Para metelas nunha máquina hai que **escoller unha reixa**: cantos píxeis, cantas medidas por segundo. E toda reixa decide, á vez, o que se conserva e o que se tira.

EN PALABRAS SINXELAS *toda conversión é unha decisión*

Non existe a conversión neutra. Cada unha destas correspondencias:

- garda **unhas relacións** (as cores veciñas, a forma da onda),
- e descarta outras (a luz exacta daquela tarde, o que pasou entre dúas medidas),
- e o descartado **non está agochado en ningures**: perdeuse.

É exactamente a imaxe do mapa e o territorio do capítulo 2. Un mapa útil non é unha reprodución perfecta; é unha representación que conserva o que che fai falta.

E agora a pregunta que abre o resto do libro.

Nos dous casos deste capítulo, o que había que conservar estaba claro: a luz en cada punto, a presión do aire en cada instante. Son cousas medíbeis e por iso a conversión de analóxico a dixital, aínda que teña perdas, é honesta e discutíbel con números.

Pero no capítulo 6 imos ter que converter en números unha cousa máis difícil de medir: **a linguaxe**. Que reixa se usa para iso? Que relacións se conservan cando se converte unha palabra en cifras? E, sobre todo, que se perde?

♥ **SEN MEDO** *non hai ningún truco pendente*

Se levas dous capítulos esperando a parte difícil, vaite decepcionar saber que xa pasou.

Ata aquí o libro só usou operacións de escola: contar, sumar, multiplicar, dividir e mirar táboas. Todo o que vén despois (as palabras convertidas en puntos, o adestramento, a atención) constrúese enriba disto e non hai ningún salto conceptual agochado polo medio. Hai máis pezas e son as mesmas pezas.

👁 **MÍRAO** *e como sabe a máquina se eses bits son unha letra ou unha cor?*

Chegados aquí aparece unha pregunta razoábel. Se todo son bits, que distingue os bits dunha foto dos dunha carta?

Nada: non hai nada nos propios bits que indique o que representan. Os mesmos oito bits, 01000001, son o número 65, a letra `A` ou un ton de gris escuro, segundo o que se faga con eles. Poden ser mesmo unha *orde* para o procesador, que é a máis rara das lecturas posíbeis e a que explica o capítulo seguinte. O sentido non está dentro: está no que a máquina ten apuntado *fóra*.

Dito con máis precisión: **o significado dos datos vén do uso**. Uns bits son unha imaxe porque hai un programa que os vai tratar como unha imaxe e son un número porque hai unha operación que os vai sumar. Fóra dese uso, non son nada en particular.

Porque a máquina leva contabilidade. Normalmente cada ficheiro vai acompañado de información sobre si mesmo: onde empeza, canto ocupa, cando se escribiu e de que tipo é. Parte diso vai no propio ficheiro, nas primeiras posicións (unha especie de etiqueta que di «isto é unha imaxe destas dimen-

sións»); parte vai nun índice separado que leva o sistema. A extensión do nome, eses `.jpg` ou `.txt`, é só a pista máis visíbel e a menos fiábel: podes cambiarlle o nome a un ficheiro e non cambia nada do que leva dentro.

E hai unha segunda cousa que matizar, porque «nos bits non hai nada» soa a que nunca se pode detectar un erro e non é así. Non se pode saber o que *son* uns bits, pero moitas veces si se pode saber que **non son o que dicían ser**, porque os datos con estrutura teñen que cumprir regras. Se ao procesador lle chega unha instrución que non existe, protesta. Se un ficheiro non trae as etiquetas que o seu formato esixe, o programa négase a abrilo. Se unha descarga non cadra coa súa cifra de control, rexéitase. Iso non é entender os datos: é comprobar que encaixan no uso que se lles quería dar.

Isto ten unha consecuencia que sorprende a moita xente. Un ordenador, en rigor, **non pode borrar**: dentro sempre hai ceros e uns, nunca hai oco. O máximo que pode facer é escribir enriba.

Así que cando borras unha foto, o habitual non é que se toquen eses bits. Chega con cambiar unha marca na contabilidade (nalgúns sistemas antigos era literalmente substituír a primeira letra do nome por un símbolo) para que o ficheiro pase da lista dos visíbeis á dos borrados. Os bytes seguen alí ata que algo os pisa. Por iso ás veces se recuperan ficheiros borrados e por iso, cando alguén quere borrar de verdade, hai que escribir enriba a propósito.

Cando ese espazo se lle acaba dando a outros datos, a información anterior desaparece. Non se foi a ningún sitio: quedou pisada por outra cousa. O único rastro que deixa é unha miga de calor.

UN CHANZO MÁIS *unha memoria USB que parece ruído*

Hai un caso que remata de deixar clara a idea de que os bits non din de que son.

Con algúns sistemas de cifrado de disco ou de memoria USB, o que queda gravado son bits que, mirados por fóra, non se distinguen de ruído puro: non hai etiquetas, non hai formato recoñecíbel, non hai nada que diga «isto é un sistema de ficheiros». E, con todo, éo. Coa chave correcta, os mesmos bits volven ser cartafoles, fotos e documentos; sen ela, a máquina non ten como saber se alí hai algo ou non hai nada.

É o exemplo extremo do que dixemos antes: o sentido non está nos bits, está no que se fai con eles.

UN SÍMBOLO NOVO *a orde dos bytes tamén é un acordo*

Un número pequeno cabe nun byte. Un número maior precisa dous ou catro. E aí aparece unha pregunta que non ten resposta natural: cal se garda primeiro?

Colle o número 4 660. En dous bytes son 18 e 52, porque o primeiro conta grupos de 256 e o segundo conta unidades:

$$18 \times 256 + 52 = 4\,608 + 52 = 4\,660$$

A máquina pode gardalos na orde 18, 52 (primeiro o que máis vale, como escribimos nós os números, empezando polos miles) ou na orde 52, 18 (primeiro o que menos vale). As dúas son igual de boas. O problema chega se quen escribe usa unha e quen le usa a outra, porque daquela o 4 660 convértese en

$$52 \times 256 + 18 = 13\,312 + 18 = 13\,330,$$

que non ten nada que ver.

Os procesadores dos ordenadores de mesa usan case todos a segunda orde; moitas redes e moitos formatos de ficheiro acordaron a primeira. Ningún dos dous bandos gañou e non fai falta que gañe: abonda con que cada un diga cal usa.

Os nomes son *big endian* e *little endian*, que veñen a ser «os do cabo gordo» e «os do cabo fino». Saíron das *Viaxes de Gulliver*: Liliput e Blefuscu andan en guerra por se os ovos se abren polo extremo groso ou polo fino. Danny Cohen colleu prestada a broma nunha nota de 1980 titulada «Sobre as guerras santas e unha petición de paz», publicada despois nunha revista [30] e o nome quedou.

Repara no que acaba de pasar, que é o mesmo do cadro anterior. Os bits non din en que orde se len, igual que non din se son unha letra ou unha cor. A orde é outro acordo que vai por fóra. Todo é convención.

O ESENCIAL DESTE CAPÍTULO

- Converter letras en números non é un descubrimento: é un **acordo**. O ASCII (1963) fixou 128 caracteres e deixou fóra case todas as linguas do mundo.
- Aquela táboa está ben deseñada: maiúscula e minúscula diferéncianse **nun só bit** e os díxitos convértense restando 48.
- **Unicode** (1991) dálle un número a cada carácter; **UTF-8** gárdaos usando entre un e catro bytes. Por iso «mañá» son catro caracteres e seis bytes.
- **Carácter e byte non son o mesmo** e os «Ã±» aparecen cando dúas partes non usan o mesmo acordo.
- Un son é unha lista de medidas tomadas moitas veces por segundo (por exemplo, 44 100); unha imaxe é unha reixa de números.
- A compresión con perdas tira o que un modelo da percepción humana di que **non se nota**. Ese modelo describe unha persoa media e alguén decidiu como é esa persoa.
- Os bits por si sós non din de que son: o tipo está apuntado fóra, nas etiquetas que leva o sistema. Borrar un ficheiro é, case sempre, marcar o seu espazo como libre.
- Nin sequera a orde na que se gardan os bytes dun número vén dada. Hai dous acordos e úsanse os dous.
- Toda conversión de analóxico a dixital conserva uns valores e descarta outros. O descartado non está agochado: desaparece.

E AGORA?

Xa hai números dentro e veñen de todas partes. Pero seguimos sen explicar como unha máquina que só suma acaba *decidindo* algo. Unha suma non decide nada e un ordenador decide constantemente. De onde sae a decisión?

Aprender

O bloque III ábrese cun verbo que en galego fai dobre traballo. Tamén a el convén devolverlle a palabra ao dicionario.

> **aprender** *v. tr.*
>
> **1.** Adquirir o coñecemento de [algo], chegar a saber por medio do estudo, a práctica, a observación, a experiencia etc. *Aprender un oficio.* Tamén absoluto: *Aprende con facilidade.*
> **2.** Reter [algo] na memoria. *Aprendeu o credo.*
> **3.** Transmitir ou facer adquirir o coñecemento de [algo]. *O profesor apréndenos cousas moi interesantes. Apréndeme a montar a cabalo.*
>
> *Dicionario da Real Academia Galega [123].*

A primeira e a terceira acepción van en sentidos contrarios: unha é de quen recibe o coñecemento e a outra, de quen o dá. Aprendemos a ler e tamén lle aprendemos a ler a alguén. Por iso o glosario traduce *train* por *aprenderlle*.

Da dobre acepción sae unha frase que outras linguas non teñen tan a man: *aprendéronllo*. Non di se o mérito é de quen ensinou ou de quen aprendeu. Os dous sentidos quedan no aire e ningún gaña.

Con estas máquinas, esa ambigüidade é xusta. Quen ensinou? En certo sentido, ninguén: o texto do que aprenden escribírono millóns de persoas que non sabían que estaban a ensinar nada (capítulo 15). E quen aprendeu? Tampouco alguén: unha morea de números que se foron movendo ata que o erro baixou (capítulo 10).

E a acepción **2**, a de reter na memoria, é xusto a que *non* lles cadra. Como vimos no capítulo 9, non lles cabe o que leron: o que queda dentro non é o texto, son as regularidades. Ninguén lle explicou nada a ninguén e case nada quedou memorizado. E, con todo, *aprendéronllo*.

Argallar

O título deste libro descansa nun verbo que en galego fai moito traballo e que non ten explicación cómoda. Pareceunos xusto rematar devolvéndolle a palabra ao dicionario, que é de onde a collemos.

> **argallar** *v.*
>
> **1.** [Persoa, especialmente neno] manipular algo para divertirse.
> **2.** Inventar mentiras ou historias.
> **3.** Facer cousas de pouca importancia ou valor.
> **4.** Concibir ou preparar algo con algunha finalidade práctica.
> **5.** Organizar ou levar a cabo algo sobre a marcha, cos medios de que se dispón no momento.
>
> *Dicionario da Real Academia Galega [123].*

> **argalhar** *v. tr.*
>
> **1.** Inventar mentiras. Mentir.
> **2.** Armar ou promover embrulhos.
> **3.** Discorrer, inventar contos ou histórias.
> **4.** Conceber um plano com uma finalidade prática.
>
> *Dicionário Estraviz [2].*

Léanse as dúas seguidas: a mesma palabra en dúas normas ortográficas do galego e o mesmo abano de sentidos, do enredo á planificación.

E de onde vén? Non se sabe con certeza. É unha resposta honesta e pouco satisfactoria. A Real Academia Galega apunta que probabelmente proceda dunha raíz prerromana [124]. Hai tamén unha proposta moito máis vella, de frei Martín Sarmiento no século XVIII, recollida entre as vinte obras que xuntou o *Dicionario de dicionarios* do Instituto da Lingua Galega [134]: emparenta *argallar* coa familia de

argadelo e *árgano*, que son os nomes dos trebellos que dan voltas para devanar ou para erguer pesos.

A proposta non está demostrada e seguramente non o estea nunca. Pero gústanos de máis para non deixala escrita nun libro coma este, porque sería unha palabra saída dos nomes das máquinas.

E agora volvamos ler as acepcións coas dezaseis leccións deste libro na cabeza.

A acepción **2** é o capítulo 14, cunha matización importante: *argallar* mentiras supón querer mentir e para querer algo fai falta alguén que queira. Dentro da máquina non hai ninguén que queira: hai unha suma de números que produce a continuación máis plausíbel. Por iso a máquina argalla **sen intención**, que é precisamente o que a fai difícil de manexar: non se lle nota, porque non está a disimular nada.

A acepción **5** é o capítulo 13 enteiro. Organizar algo sobre a marcha, cos medios de que se dispón no momento: os medios son a xanela de contexto e o momento é cada palabra.

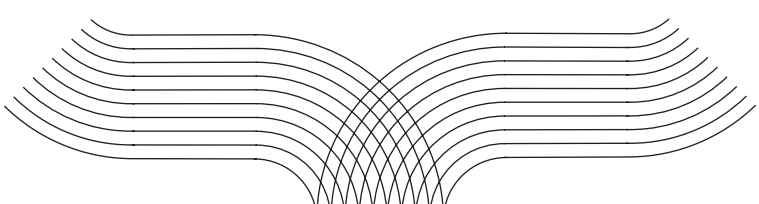
A acepción **3** é o que fan estas ferramentas a maior parte do tempo e non pasa nada. Moitas das cousas que lles pedimos son de pouca importancia e a alguén lle levarían a tarde.

A acepción **4** segue sendo nosa. Concibir algo cunha finalidade práctica esixe ter unha finalidade e as finalidades non se calculan: é o capítulo 16.

E a acepción **1**, a de quen, sobre todo de neno, manipula algo só para divertirse, é o motivo deste libro. Todo empezou nunha aula, cunha pregunta sobre unha calculadora que ninguén se fixera nunca. Argallar, no mellor sentido da palabra, é o que fixeron Verea cos seus prismas de latón, Shannon cos seus interruptores e quen queira que, un día, se pare a preguntarse como raio é posíbel que a máquina que só podía sumar responda.

A máquina que só podía sumar xa é quen de argallar.

En que acepción, decídelo ti.





Iván Area Carracedo

Iván Area Carracedo é catedrático de universidade de Matemática Aplicada na Escola de Enxeñaría Aeronáutica e do Espazo da Universidade de Vigo, no campus de Ourense. É investigador do IFCAE, ten publicado máis de 130 artigos en revistas internacionais de recoñecido prestixio e dirixido seis teses de doutoramento. É membro do grupo de innovación docente IADA. Foi secretario xeral do CIMPA entre 2016 e 2020, vicerreitor de Relacións Institucionais entre 2006 e 2010, director de Avaliación de Profesorado da ANECA en 2020 e delegado do reitor na Comisión Interuniversitaria de Galicia de 2018 a 2020 e de 2022 a 2026. Recibiu distintos premios polo seu labor docente. En O Perípato (https://area.webs.uvigo.gal/o-peripato/) publica reflexións sobre pedagoxía, avaliación, universidade, ciencia e sociedade.



Manuel Caeiro Rodríguez

Manuel Caeiro Rodríguez é catedrático de universidade de Enxeñaría Telemática na Escola de Enxeñaría de Telecomunicación da Universidade de Vigo e investigador de atlanTTic. Coordina o Máster Inter-Universitario en Ciberseguridade impartido conxuntamente entre as universidades de Vigo e da Coruña. A súa investigación céntrase nas tecnoloxías educativas: contornos de formación en liña, analítica da aprendizaxe, aprendizaxe autorregulada, recursos educativos abertos, realidade virtual e, nos últimos anos, os usos da intelixencia artificial xenerativa na docencia. En 2024 coordinou o grupo de máis de vinte docentes que elaborou as dúas guías pioneiras da súa escola sobre estas ferramentas nas aulas universitarias, unha dirixida ao profesorado e outra ao estudantado.



Arno Formella

Arno Formella é alemán e profesor do Departamento de Informática da Universidade de Vigo, na área de Linguaxes e Sistemas Informáticos. Foi o primeiro director da Escola de Enxeñaría Aeronáutica e do Espazo. Dirixe actualmente a Escola Superior de Enxeñaría Informática. Coordinaba os programas de doutoramento Tecnoloxías Avanzadas para o Desenvolvemento de Software Intelixente (2007-2015) e Sistemas de Software Intelixentes e Adaptables (2013-2026). O seu traballo é o da informática aplicada a problemas doutras disciplinas, en particular na parte algorítmica. É responsable do grupo de investigación Laboratorio de Intelixencia Artificial Aplicada (LIA2) e investigador do IFCAE.

Miscelánea

Serie de textos misceláneos

Últimas publicacións na colección

English Pronunciation is Fun (e-ProFun). Un juego serio para enseñar y aprender la pronunciación del inglés en la educación primaria y secundaria. (2026)
María de los Ángeles Gómez González, José Carlos López Ardao, Alfonso Lago Ferreiro, Fernando Fraga Varela

Aprendizaxe cooperativa na universidade e en diferentes contextos. Voces dende a experiencia. (2025)
Almudena Alonso-Ferreiro, Isabel Mociño-González, Breogán Riobóo-Lois, María Ainoa Zabalza-Cerdeiriña

Los sonidos del inglés en el Camino de Santiago (e-SoundWay). Un juego serio para aprender la pronunciación inglesa. (2025)
María de los Ángeles Gómez González, Alfonso Lago Ferreiro, Milagros Torrado Cespón, José María Díaz Lage

La forma que reconstruye. Sociología de los espacios fílmicos de Madrid (1958-2021). (2025)
Eva Blanco Alonso

Voces de mujeres. 100 voces, 100 vidas, 100 historias compartidas. (2025)
María Méndez Santos, Ana Varela Suárez, Susana Rodríguez Barcia

9 788411 881326

A máquina que só podía sumar xa é quen de argallar

Como funcionan por dentro as intelixencias artificiais

Un día calquera, nunha aula, alguén pregunta: como fai a calculadora do teléfono para darche a raíz cadrada de dous nun pestanexo, se dentro do aparello só hai sumas? O libro empeza por aí e explica o mecanismo que o fai posíbel de xeito sinxelo, sen trampa nin cartón.

Como adoita pasar, cada resposta abre novas preguntas. Como entran na máquina as letras, as cores e o son? Como se converte unha palabra nun punto dun mapa que ninguén debuxou? Que significa «aprender» cando o único que cambia son números? Que é a atención, a peza que en 2017 o cambiou todo? E por que estes programas mencionan libros e artigos que nunca se escribiron coa mesma seguridade que os existentes?

O libro responde a estas preguntas sen saltos e sen dar nada por sabido, ata chegar ás intelixencias artificiais que escriben, conversan e, ás veces, inventan. Non fai falta saber programar nin lembrar as matemáticas do instituto: abonda con saber sumar, restar, multiplicar e dividir. E con ter curiosidade. Cada conta pódese facer a man na marxe da páxina. Os cadros de cores permiten lelo a varias velocidades, desde a idea en palabras sinxelas ata o chanzo máis técnico. Máis de trinta exercicios, cada un coa súa resposta razoada, convidan a comprobalo pola túa conta.

Escrito e pensado orixinalmente en galego, o libro mide tamén o que lle custa á nosa lingua este novo xeito de ler e escribir. E explica por que cada texto público en galego conta.

Ao final, a máquina non resulta máxica nin pensante, senón algo máis interesante: unha montaña de operacións moi simples que produce algo que se parece moito a pensar sen selo. E queda unha pregunta que xa podes responder ti: como é quen de argallar todo isto?

Servizo de Publicacións
Universidade Vigo