

ÍNDICE

Prólogo	1
Acerca del libro	5
PARTE I	
INTRODUCCIÓN A LA COMPUTACIÓN Y LA PROGRAMACIÓN	9
Capítulo 1. ¿Qué es la computación?	11
1.1 Modelos de computación.....	13
1.1.1 Máquina de Turing.....	14
1.1.2 Cálculo lambda.....	20
1.1.3 Otros	23
1.2 Tesis de Church-Turing	23
1.2.1 Implicaciones filosóficas.....	24
1.3 Filosofía de la ciencia de la computación	25
Capítulo 2. ¿Qué es la programación?	27
2.1 Algoritmos	28
2.2 Especificación.....	29
2.2.1 Verificación formal.....	29
2.2.2 ¿Pensar antes de programar?.....	34
2.3 Implementación.....	36
Capítulo 3. Lenguajes de programación	39
3.1 Características de los lenguajes de programación	40
3.2 Paradigmas clásicos de la programación.....	43
3.2.1 Programación imperativa.....	44
3.2.2 Programación orientada a objetos	46
3.2.3 Programación lógica	51
3.2.4 Programación funcional.....	53

PARTE II	
CÁLCULO LAMBDA	55
Capítulo 4. ¿Qué es el cálculo lambda?	57
4.1 Historia.....	58
4.2 Sintaxis	60
4.2.1 Notación Backus-Naur extendida	61
4.2.2 Cálculo lambda.....	67
Capítulo 5. Operadores y variables	73
5.1 Operadores.....	74
5.1.1 Abstracción	75
5.1.2 Aplicación	80
5.2 Variables	83
5.2.1 Bound	83
5.2.2 Free	85
Capítulo 6. Reducción	89
6.1 Reducción alfa (α)	90
6.2 Reducción beta (β).....	91
6.2.1 Reglas	92
6.2.2 Teorema de Church-Rosser	93
6.3 Reducción eta (η).....	96
Capítulo 7. Aritmética	99
7.1 Números.....	100
7.2 Operaciones	101
7.2.1 Sucesor	101
7.2.2 Suma.....	103
7.2.3 Multiplicación	104
7.2.4 Predecesor.....	106
7.2.5 Resta	108
Capítulo 8. Condicionales	111
8.1 Valor booleano.....	112
8.2 Operadores.....	113
8.2.1 AND	113
8.2.2 OR.....	115
8.2.3 NOT	116
8.2.4 XOR.....	117

Capítulo 9. Tuplas y listas	121
9.1 Tuplas	123
9.1.1 Operaciones de acceso	124
9.2 Listas	125
9.2.1 Append.....	125
9.2.2 Head	127
9.2.3 Tail	128
9.2.4 IsEmpty.....	128
Capítulo 10. Tipos	131
10.1 Cálculo- λ tipado.....	132
10.2 Definiciones de reglas	133
10.2.1 Variable	133
10.2.2 Abstracción	134
10.2.3 Aplicación.....	134
10.3 Reducción de tipo	135
10.4 Una breve introducción a Haskell	137
10.4.1 Funciones	138
10.4.2 Listas.....	141
10.4.3 Tuplas.....	144
10.5 Otras características.....	146
10.5.1 Pattern matching.....	146
10.5.2 Guards.....	148
Capítulo 11. Cálculo-λ como base de un lenguaje de programación real	151
11.1 Diferencias e influencias.....	152
11.1.1 Lenguajes de programación funcional.....	155
11.2 Límites del cálculo- λ	159
PARTE III	
PROGRAMACIÓN FUNCIONAL	161
Capítulo 12. ¿Qué es la programación funcional?	163
12.1 Introducción	166
12.1.1 Justificaciones previas	166
12.1.2 Racket	168
12.1.3 Python	169
12.2 Función, recursión y datos	170
12.2.1 Sobre funciones	172
12.2.2 Recursividad.....	175
12.2.3 Lista.....	181

12.3	Principales conceptos de la programación funcional.....	184
12.3.1	Funciones puras	185
12.3.2	Higher-order functions.....	187
12.3.3	Pattern matching.....	192
12.3.4	Lazy evaluation.....	196
12.3.5	Transparencia referencial.....	199
12.3.6	Inmutabilidad	203
Capítulo 13.	Estructuras de datos	205
13.1	Lista	206
13.1.1	Búsqueda	211
13.1.2	Inserción.....	212
13.1.3	Eliminación.....	213
13.1.4	Filtrado	213
13.2	Tabla hash	214
13.2.1	Búsqueda	217
13.2.2	Inserción.....	218
13.2.3	Eliminación.....	218
13.3	Par	219
13.3.1	Operadores de acceso.....	221
13.4	Estructura de tipos.....	222
13.4.1	Operadores de acceso.....	224
13.5	Árbol de búsqueda binario.....	225
13.5.1	Búsqueda	229
13.5.2	Cantidad de elementos.....	231
Capítulo 14.	Algoritmos	233
14.1	Ordenamiento	236
14.1.1	Quicksort.....	237
14.1.2	Merge sort	240
14.2	Recursividad	244
14.2.1	Torre de Hanói.....	245
14.3	Búsqueda de subcadenas.....	248
14.3.1	Karp-Rabin	250
14.4	Compresión de datos.....	255
14.4.1	Codificación Huffman	256
Capítulo 15.	Crear un pequeño lenguaje de programación usando Racket.....	265
15.1	Especificación.....	265
15.2	Analizador léxico.....	269
15.3	Analizador sintáctico.....	272

15.4 Intérprete.....	273
15.4.1 Pruebas	277
Epílogo - Lecturas recomendadas.....	281
Agradecimientos.....	283
Apéndice A - Notación Big O.....	285
Apéndice B - Introducción a TLA+ (PlusCal).....	287
Bibliografía	293
Glosario.....	295
Índice alfabético.....	299

BIBLIOGRAFÍA

- Abelson, H.; Sussman, G.J.; Sussman, J. (1996). *Structure and Interpretation of Computer Program* (2 ed.). MIT Press.
- Aho, A.V.; Lam, M.S.; Sethi, R.; Ullman, J.D. (2006). *Compilers: Principles, Techniques, and Tools*. Pearson.
- Ambler, A.L.; Burnett, M.M.; Zimmerman, B. A. (1992). «Operational Versus Definitional: A Perspective on Programming Paradigms». *Computer* 25, 28-43.
- Barendregt, H. (1999). «The impact of the lambda calculus in logic and computer science». *The Bulletin of Symbolic Logic* 3(2).
- Barendregt, H.; Barendsen, E. (1994). *Introduction to Lambda Calculus*.
- Barendsen, E. (2006). *Introduction to Type Theory*.
- Butterfield, A.; Ekembe Ngondi, G. (eds.) (2016). *A Dictionary of Computer Science*. Oxford University Press.
- Church, A. (1936). «An unsolvable problem of elementary number theory».
- Dasgupta, S. (2016). *Computer Science - A very short introduction*. Oxford.
- Davis, M. (2000). *The Universal Computer: The Road from Leibniz to Turing*. W. W. Norton & Company.
- Davis, M. (2004). «The Myth of Hypercomputation». En: *Alan Turing: Life and Legacy of a Great Thinker*, 195-211.
- Denning, P.J.; Freeman, P.A. (2009). «Computing's Paradigm». *Communications of the ACM*, 28-30.
- Denning, P.J.; Tedre, M. (2019). *Computational thinking*. MIT Press Essential Knowledge series.
- Gabbrielli, M.; Martini, S. (2010). *Programming Languages: Principles and Paradigms*. Springer.
- Geruzzi, P.E. (2012). *Computing. A concise history*. MIT Press Essential Knowledge series.

- Gurevich, Y. (2012). «What is an Algorithm?». *Technical Report MSR-TR-2011-116*.
- Hindley, R. (2006). History of lambda-calculus and combinatory logic.
- Knuth, D. (1998). *The Art of Computer Programming. Sorting and Searching Vol. 3*. Addison Wesley.
- Lämmel, R. (2018). «An Excursion into the Lambda Calculus». En: *Software Languages. Syntax, Semantics, and Metaprogramming*. Springer.
- Lampert, L. (2015). «Who builds a house without drawing blueprints?» En: *Communications of the ACM* 58(4), 38-41.
- Lampert, L. (2018). «If You're Not Writing a Program, Don't Use a Programming Language». En: *Distributed Computing & Education Column*.
- Lucas Alba, S. (2019). *La máquina que cambió el mundo: Génesis, desarrollo y evolución del ordenador*. Complutense ediciones.
- Massalõgin, V. (2008). *Visual Lambda Calculus*. Tesis de máster en University of Tartu.
- Okasaki, C. (2008). *Purely Functional Data Structures* (2 ed.) Cambridge University Press.
- Révész, G. E. (1988). *Lambda-calculus, Combinators and Functional Programming*. Cambridge University Press.
- Rosser, B. (1984). «Highlights of the history of the lambda calculus». En: *IEEE Annals of the History of Computing* 6, 337-349.
- Sipser, M. (1997). *Introduction to the Theory of Computation*. PWS publishing company.
- Thompson, S. (1991). *Type Theory and Functional Programming*. International Computer Science Series. Addison-Wesley.
- Trakhtenbrot, B. A. (1960). «Algorithms».
- Turing, A. (1936). «On Computable Numbers, with an Application to the Entscheidungsproblem».
- Turing, A. (2012). *¿Puede pensar una máquina?* KRK ediciones.
- Turner, R. (2018). *Computational Artifacts. Towards a Philosophy of Computer Science*. Springer.
- von Neumann, J. (1945). *First Draft of a Report on EDVAC*. University of Pennsylvania.
- Wirth, N. (1976). *Algorithms + Data Structures = Programs*. Prentice Hall.