

TABLA DE CONTENIDO

INTRODUCCIÓN	15
Acerca de este libro.....	16
Organización del libro	17
Convenciones de formato	18
Acerca del autor.....	19
CAPÍTULO 1. INSTALACIÓN Y USO DE GO	21
1.1 Instalando Go	21
1.2 Comandos básicos de Go	22
1.3 Editando su código en Go.....	23
1.4 Compilando y ejecutando su primer programa.....	24
CAPÍTULO 2. SINTAXIS BÁSICA DE GO	27
2.1 Tipos de datos básicos.....	27
2.2 Cadenas de texto. El tipo <code>string</code>	28
2.3 Definición de variables.....	29
2.4 Conversiones explícitas de tipos	30
2.5 Constantes.....	30
2.6 Bases de numeración.....	31
2.7 Operadores numéricos.....	32
2.8 Operadores numéricos de comparación.....	33
2.9 Operadores aplicables al tipo <code>string</code>	34
2.10 Operadores lógicos con <code>bool</code>	34

2.11	Operadores lógicos a nivel de bit.....	35
2.12	Salida estándar de datos.....	35
2.13	Entrada estándar de datos	37
CAPÍTULO 3.	CONTROL DE FLUJO	39
3.1	Bloques condicionales	39
3.1.1	if.....	39
3.1.2	if ... else.....	40
3.1.3	switch - case.....	42
3.2	Órdenes iterativas (bucles for).....	45
3.3	Contexto y ocultación de variables	48
CAPÍTULO 4.	APUNTADORES	49
4.1	Definición de un apuntador.....	49
4.2	La referencia a nil.....	50
4.3	Apuntando hacia una variable	50
4.4	Leyendo o modificando el valor apuntado.....	51
4.5	Valores versus referencias	52
CAPÍTULO 5.	FUNCIONES	55
5.1	Definición e invocación.....	55
5.2	Retorno de valores.....	57
5.3	Retorno de múltiples valores	58
5.4	Retorno de múltiples valores nombrados.....	58
5.5	El identificador vacío	59
5.6	Paso por valor vs. paso por referencia.....	59
5.7	Literales de función	61
5.8	Otras consideraciones	63
CAPÍTULO 6.	ESTRUCTURAS DE DATOS LINEALES	65
6.1	Vectores.....	65
6.2	Porciones	67
6.3	Declarando variables a porciones.....	68
6.4	Añadir elementos a una porción. Función append.....	69
6.5	Medir dimensiones con len y cap	70
6.6	Controlar el tamaño inicial con make	71

6.7	Copia de porciones con copy	72
6.8	Uso de porciones en funciones	72
6.9	Recorriendo vectores y porciones.....	74
6.10	Creando "vistas" desde las porciones.....	75
6.11	Funciones con número variable de argumentos	77
6.12	El operador difusor.....	77
CAPÍTULO 7. CADENAS DE TEXTO		79
7.1	Diferencias con porciones y vectores.....	80
7.2	Obteniendo la longitud de un string.....	81
7.3	De string a porción.....	82
7.4	Construcción dinámica de cadenas.....	83
7.4.1	Concatenación de cadenas.....	84
7.4.2	Construcción con strings.Builder	84
7.4.3	Paquete fmt.....	85
CAPÍTULO 8. DICCIONARIOS (MAPAS)		87
8.1	Declaración de mapas.....	87
8.2	Acceso a elementos.....	88
8.3	Eliminando entradas con delete.....	89
8.4	Recorriendo mapas con range.....	89
8.5	Contando el número de elementos.....	90
8.6	Conjuntos.....	91
8.7	Detalles internos de map.....	92
CAPÍTULO 9. ORGANIZACIÓN DE CÓDIGO		95
Paquetes y módulos		
9.1	Paquetes (package).....	95
9.2	Módulos	96
9.3	Creando módulos y paquetes.....	98
9.4	Importando paquetes del módulo local	99
9.4.1	Dependencias circulares	100
9.5	Incorporando paquetes de módulos externos.....	100
9.6	Copias locales de módulos. El directorio vendor	102
9.7	Elementos públicos y privados a nivel de paquete	103

9.8	Alias de paquete.....	105
9.9	La función <code>init</code>	106
CAPÍTULO 10.	DEFINICIÓN DE TIPOS DE DATOS	109
10.1	Tipos a partir de porciones.....	110
10.2	Tipos a partir de mapas.....	111
10.3	Tipos funcionales.....	112
10.4	Receptores de función. Métodos.....	115
10.5	Tipos pseudoenumerados.....	118
10.5.1	El operador <code>iota</code>	119
10.6	Caso de estudio: <code>time.Duration</code>	122
CAPÍTULO 11.	TIPOS DE DATOS ESTRUCTURADOS	125
	<code>Struct</code>	
11.1	Tipos de datos estructurados: <code>struct</code>	125
11.2	Punteros a <code>struct</code>	127
11.3	Receptores de función y creación de métodos.....	128
11.4	Incrustado de estructuras.....	128
11.5	La estructura vacía: <code>struct{}</code>	131
11.6	Caso práctico: opciones funcionales como alternativa a constructores.....	132
CAPÍTULO 12.	INTERFACES	137
12.1	Caso de estudio: la interfaz <code>Stringer</code>	139
12.2	La filosofía del tipado estructural.....	139
12.3	Implementando interfaces: receptores ¿mediante apuntadores o mediante valores?.....	140
12.4	La interfaz vacía <code>interface{}</code>	142
12.5	Manejo seguro de tipos de datos.....	143
12.6	Incrustando interfaces.....	146
CAPÍTULO 13.	GESTIÓN DE ERRORES	147
13.1	La interfaz <code>error</code>	147
13.2	Instanciando errores de manera genérica.....	148
13.3	Comprobación de tipos de error.....	148
13.3.1	Errores centinela.....	149

13.3.2 Distintas implementaciones de error	152
13.4 Envolviendo errores.....	154
13.5 Verificando la cadena de errores: <code>errors.As</code>	156
13.6 <code>defer</code>	158
13.7 Entrando en pánico	160
13.8 Función <code>panic</code>	161
13.9 Función <code>recover</code>	161
CAPÍTULO 14. ENTRADA Y SALIDA	165
Flujos de datos	
14.1 Interfaces <code>io.Writer</code> e <code>io.Reader</code>	165
14.2 Archivos de disco	167
14.3 Entrada y salida formateada.....	169
14.4 Paquete <code>bufio</code>	171
14.5 Paquete <code>ioutil</code>	172
CAPÍTULO 15. PARALELISMO Y CONCURRENCIA	175
Gorrutinas	
15.1 Un poco de historia	175
15.2 Gorrutinas.....	176
15.3 Sincronización mediante <code>sync.WaitGroup</code>	180
15.4 Problemas de concurrencia: condiciones de carrera	182
15.5 Sincronización mediante <code>sync.Mutex</code>	186
15.5.1 <code>sync.RWMutex</code>	188
15.6 Sincronización mediante <code>atomic</code>	188
15.7 Conclusiones: ¿cuándo y cómo sincronizar gorrutinas?	189
CAPÍTULO 16. CANALES	191
16.1 Creación, uso y cierre.....	191
16.2 Canales solo de lectura y de escritura.....	193
16.3 Bloqueo en la escritura: canales con o sin búfer	195
16.4 Iterando canales con <code>for</code>	197
16.5 Múltiples receptores.....	198
16.6 Sincronización mediante canales	199
16.7 Demultiplexión con <code>select</code>	201

16.8	Cancelando lecturas después de un tiempo de espera.....	202
16.9	Cancelando tareas mediante contextos	204
CAPÍTULO 17.	SERVICIOS WEB	207
17.1	HTTP explicado en 3 minutos.....	207
17.2	REST explicado en 3 minutos	209
17.3	Creación de un servicio HTTP en Go	210
17.3.1	Interfaz <code>http.Handler</code>	211
17.3.2	Funciones <code>http.ListenAndServe</code> y <code>http.ListenAndServeTLS</code>	211
17.3.3	Ejemplo de servidor HTTP	212
17.4	Creación de un cliente HTTP en Go	213
17.4.1	Ejemplo de cliente HTTP.....	214
17.5	Ejemplo práctico de servicio REST.....	215
17.5.1	Probando el servicio REST.....	220
CAPÍTULO 18.	SERIALIZACIÓN DE DATOS	223
18.1	Serialización de tipos Go a JSON	224
18.2	Deserialización de JSON a tipos Go.....	225
18.3	Serializando y deserializando documentos JSON sin formato.....	227
18.4	Serialización de porciones y arrays.....	227
18.5	Serialización y deserialización en otros formatos.....	228
CAPÍTULO 19.	CONEXIÓN A BASES DE DATOS SQL	231
19.1	Carga de controlador	232
19.2	Abriendo una base de datos	233
19.3	Modificando la base de datos	233
19.4	Consultando datos.....	234
19.5	Declaraciones preparadas	235
19.6	Transacciones	237
19.7	Reserva de conexiones.....	238
CAPÍTULO 20.	PRUEBAS AUTOMATIZADAS DE SOFTWARE	241
20.1	Código a probar: la función Factorial	241
20.2	El paquete <code>testing</code>	242
20.3	Probando servicios HTTP	244

20.4	Pruebas de rendimiento.....	246
20.5	Cobertura de las pruebas	247
APÉNDICE.	PROGRAMACIÓN GENÉRICA	249
	Tipos de datos genéricos.....	252
	Valor cero de un tipo genérico	254
	Funciones genéricas	254
	Restricciones sobre argumentos de tipo	255
	Restricciones como interfaces.....	258
	Definiendo nuevas restricciones.....	259
	Múltiples argumentos de tipo	261
	Ejemplo: MultiMapa	261