

TABLA DE CONTENIDO

INTRODUCCIÓN	13
Acerca de este libro.....	14
Organización del libro	15
Convenciones de formato	16
Acerca del autor.....	17
CAPÍTULO 1. INSTALACIÓN Y USO DE GO	19
1.1 Instalando Go	19
1.2 Comandos básicos de Go	20
1.3 Editando su código en Go.....	21
1.4 Compilando y ejecutando su primer programa.....	22
CAPÍTULO 2. SINTAXIS BÁSICA DE GO	25
2.1 Tipos de datos básicos.....	25
2.2 Cadenas de texto. El tipo <code>string</code>	26
2.3 Definición de variables.....	27
2.4 Conversiones explícitas de tipos	28
2.5 Constantes.....	28
2.6 Bases de numeración.....	29
2.7 Operadores numéricos.....	30
2.8 Operadores numéricos de comparación.....	31
2.9 Operadores aplicables al tipo <code>string</code>	31
2.10 Operadores lógicos con <code>bool</code>	32

2.11	Operadores lógicos a nivel de bit.....	32
2.12	Salida estándar de datos.....	33
2.13	Entrada estándar de datos	35
CAPÍTULO 3.	CONTROL DE FLUJO	37
3.1	Bloques condicionales	37
3.1.1	if.....	37
3.1.2	if ... else.....	38
3.1.3	switch - case.....	40
3.2	Órdenes iterativas (bucles for).....	43
3.3	Contexto y ocultación de variables	46
CAPÍTULO 4.	APUNTADORES	47
4.1	Definición de un apuntador.....	47
4.2	La referencia a nil.....	48
4.3	Apuntando hacia una variable	48
4.4	Leyendo o modificando el valor apuntado.....	49
4.5	Valores versus referencias	50
CAPÍTULO 5.	FUNCIONES	53
5.1	Definición e invocación.....	53
5.2	Retorno de valores.....	55
5.3	Retorno de múltiples valores	56
5.4	Retorno de múltiples valores nombrados.....	56
5.5	El identificador vacío	57
5.6	Paso por valor vs. paso por referencia.....	57
5.7	Literales de función	59
5.8	Otras consideraciones	61
CAPÍTULO 6.	ESTRUCTURAS DE DATOS LINEALES	63
6.1	Vectores.....	63
6.2	Porciones	65
6.3	Declarando variables a porciones.....	66
6.4	Añadir elementos a una porción. Función append.....	67
6.5	Medir dimensiones con len y cap	68
6.6	Controlar el tamaño inicial con make	69

6.7	Copia de porciones con copy	70
6.8	Uso de porciones en funciones	70
6.9	Recorriendo vectores y porciones.....	72
6.10	Creando "vistas" desde las porciones.....	73
6.11	Funciones con número variable de argumentos	75
6.12	El operador difusor.....	75
CAPÍTULO 7. CADENAS DE TEXTO		77
7.1	Diferencias con porciones y vectores.....	78
7.2	Obteniendo la longitud de un string.....	79
7.3	De string a porción.....	80
7.4	Construcción dinámica de cadenas.....	81
7.4.1	Concatenación de cadenas.....	82
7.4.2	Construcción con strings.Builder	82
7.4.3	Paquete fmt.....	83
CAPÍTULO 8. DICCIONARIOS (MAPAS)		85
8.1	Declaración de mapas.....	85
8.2	Acceso a elementos.....	86
8.3	Eliminando entradas con delete.....	87
8.4	Recorriendo mapas con range.....	87
8.5	Contando el número de elementos.....	88
8.6	Conjuntos.....	89
8.7	Detalles internos de map.....	90
CAPÍTULO 9. ORGANIZACIÓN DE CÓDIGO		93
Paquetes y módulos		
9.1	Paquetes (package).....	93
9.2	Módulos	94
9.3	Creando módulos y paquetes.....	96
9.4	Importando paquetes del módulo local	97
9.4.1	Dependencias circulares	98
9.5	Incorporando paquetes de módulos externos.....	98
9.6	Copias locales de módulos. El directorio vendor	100
9.7	Elementos públicos y privados a nivel de paquete	101

9.8	Alias de paquete.....	103
9.9	La función <code>init</code>	104
CAPÍTULO 10.	DEFINICIÓN DE TIPOS DE DATOS	107
10.1	Tipos a partir de porciones.....	108
10.2	Tipos a partir de mapas.....	109
10.3	Tipos funcionales.....	110
10.4	Receptores de función. Métodos.....	113
10.5	Tipos pseudoenumerados.....	116
10.5.1	El operador <code>iota</code>	117
10.6	Caso de estudio: <code>time.Duration</code>	120
CAPÍTULO 11.	TIPOS DE DATOS ESTRUCTURADOS	123
	<code>Struct</code>	
11.1	Tipos de datos estructurados: <code>struct</code>	123
11.2	Punteros a <code>struct</code>	125
11.3	Receptores de función y creación de métodos.....	126
11.4	Incrustado de estructuras.....	126
11.5	La estructura vacía: <code>struct{}</code>	129
11.6	Caso práctico: opciones funcionales como alternativa a constructores.....	130
CAPÍTULO 12.	INTERFACES	135
12.1	Caso de estudio: la interfaz <code>Stringer</code>	137
12.2	La filosofía del tipado estructural.....	137
12.3	Implementando interfaces: receptores ¿mediante apuntadores o mediante valores?.....	138
12.4	La interfaz vacía <code>interface{}</code>	140
12.5	Manejo seguro de tipos de datos.....	140
12.6	Incrustando interfaces.....	144
CAPÍTULO 13.	GESTIÓN DE ERRORES	145
13.1	La interfaz <code>error</code>	145
13.2	Instanciando errores de manera genérica.....	146
13.3	Comprobación de tipos de error.....	146
13.3.1	Errores centinela.....	147

13.3.2 Distintas implementaciones de error	150
13.4 Envolviendo errores.....	152
13.5 Verificando la cadena de errores: <code>errors.As</code>	154
13.6 <code>defer</code>	156
13.7 Entrando en pánico	158
13.8 Función <code>panic</code>	159
13.9 Función <code>recover</code>	159
CAPÍTULO 14. ENTRADA Y SALIDA	163
Flujos de datos	
14.1 Interfaces <code>io.Writer</code> e <code>io.Reader</code>	163
14.2 Archivos de disco	165
14.3 Entrada y salida formateada.....	167
14.4 Paquete <code>bufio</code>	169
14.5 Paquete <code>ioutil</code>	170
CAPÍTULO 15. PARALELISMO Y CONCURRENCIA	173
Gorrutinas	
15.1 Un poco de historia	173
15.2 Gorrutinas.....	174
15.3 Sincronización mediante <code>sync.WaitGroup</code>	178
15.4 Problemas de concurrencia: condiciones de carrera	180
15.5 Sincronización mediante <code>sync.Mutex</code>	184
15.5.1 <code>sync.RWMutex</code>	186
15.6 Sincronización mediante <code>atomic</code>	186
15.7 Conclusiones: ¿cuándo y cómo sincronizar gorrutinas?	187
CAPÍTULO 16. CANALES	189
16.1 Creación, uso y cierre.....	189
16.2 Canales solo de lectura y de escritura.....	191
16.3 Bloqueo en la escritura: canales con o sin búfer	193
16.4 Iterando canales con <code>for</code>	195
16.5 Múltiples receptores.....	196
16.6 Sincronización mediante canales	197
16.7 Demultiplexión con <code>select</code>	199

16.8	Cancelando lecturas después de un tiempo de espera.....	200
16.9	Cancelando tareas mediante contextos	202
CAPÍTULO 17.	SERVICIOS WEB	205
17.1	HTTP explicado en 3 minutos.....	205
17.2	REST explicado en 3 minutos	207
17.3	Creación de un servicio HTTP en Go	208
17.3.1	Interfaz <code>http.Handler</code>	209
17.3.2	Funciones <code>http.ListenAndServe</code> y <code>http.ListenAndServeTLS</code>	209
17.3.3	Ejemplo de servidor HTTP	210
17.4	Creación de un cliente HTTP en Go	211
17.4.1	Ejemplo de cliente HTTP.....	212
17.5	Ejemplo práctico de servicio REST.....	213
17.5.1	Probando el servicio REST.....	218
CAPÍTULO 18.	SERIALIZACIÓN DE DATOS	221
18.1	Serialización de tipos Go a JSON	222
18.2	Deserialización de JSON a tipos Go.....	223
18.3	Serializando y deserializando documentos JSON sin formato.....	225
18.4	Serialización de porciones y arrays.....	225
18.5	Serialización y deserialización en otros formatos.....	226
CAPÍTULO 19.	CONEXIÓN A BASES DE DATOS SQL	229
19.1	Carga de controlador	230
19.2	Abriendo una base de datos	231
19.3	Modificando la base de datos	231
19.4	Consultando datos.....	232
19.5	Declaraciones preparadas	233
19.6	Transacciones	235
19.7	Reserva de conexiones.....	236
CAPÍTULO 20.	PRUEBAS AUTOMATIZADAS DE SOFTWARE	239
20.1	Código a probar: la función Factorial	239
20.2	El paquete <code>testing</code>	240
20.3	Probando servicios HTTP	242
20.4	Pruebas de rendimiento.....	244
20.5	Cobertura de las pruebas	245