

Contenido

Prefacio	XI
----------------	----

PARTE 1

RESUMEN DEL LENGUAJE

JAVASCRIPT Y NODE.JS	1
----------------------------	---

CAPÍTULO 1

Introducción a Node.js	3
------------------------------	---

1.1. ¿Por qué es Node.js tan popular?.....	4
--	---

1.1.1. Liger y rápido	4
-----------------------------	---

1.1.2. Multiplataforma y multipropósito.....	5
--	---

1.1.3. Curva de aprendizaje fácil	5
---	---

1.1.4. Ecosistema.....	6
------------------------	---

1.1.5. Impulsado por la comunidad	7
---	---

1.2. La arquitectura de un solo hilo de Node.js.....	8
--	---

1.2.1. La E/S necesita hacerse de manera diferente ...	8
--	---

1.2.2. E/S no bloqueante	9
--------------------------------	---

1.2.3. Profundizando en el bucle de eventos	11
---	----

1.3. Versiones de Node.js	12
---------------------------------	----

1.3.1. Versionado semántico (SemVer).....	12
---	----

1.3.2. Detalles de la versión.....	13
------------------------------------	----

1.3.3. Calendario de lanzamientos.....	15
--	----

1.4. Resumen	17
--------------------	----

1.4.1. Lectura adicional	18
--------------------------------	----

CAPÍTULO 2

Configurando el entorno de desarrollo	19
---	----

2.1. Requisitos técnicos	20
--------------------------------	----

2.2. Instalando Node.js en macOS, Windows y Linux	20
---	----

2.2.1. macOS	21
--------------------	----

2.2.2. Windows	22
----------------------	----

2.2.3. Linux.....	22
-------------------	----

2.2.4. Otros entornos	23
-----------------------------	----

2.2.5. Verificando la instalación	23
---	----

2.3. Gestionando las versiones de Node.js.....	24
--	----

2.4. Administrando Node.js usando nvm	25
---	----

2.4.1. Instalación y uso de versiones	26
---	----

2.4.2. Usando un archivo .nvmrc	27
---------------------------------------	----

2.5. REPL de Node.js	29
----------------------------	----

2.6. Interactuando con JavaScript usando Chrome DevTools.....	31
---	----

2.6.1. El panel Console	32
-------------------------------	----

2.6.2. El panel Network.....	32
------------------------------	----

2.6.3. Usando Chrome DevTools	33
-------------------------------------	----

2.7. Resumen	34
--------------------	----

2.7.1. Lectura adicional	35
--------------------------------	----

CAPÍTULO 3**Fundamentos de JavaScript 37**

3.1. Requisitos técnicos 38

3.2. JavaScript es un lenguaje poderoso 38

3.3. Entendiendo la versión TC39 40

3.3.1. Entendiendo la versión ECMAScript 40

3.3.2. ¿Qué se incluirá en la próxima versión de JavaScript? .. 41

3.3.3. ¿Qué no se incluye en la especificación de JavaScript? 41

3.4. Explorando la documentación de JavaScript 42

3.5. *Linting* del código JavaScript 433.6. Comentando el código JavaScript 44
3.6.1. Uso de JSDoc 45

3.7. Imprimiendo valores y depurando de una forma ágil..... 46

3.8. Variables y constantes 47
3.8.1. Convenciones de nomenclatura..... 48

3.8.2. let vs const 48

3.9. Entendiendo los tipos de datos 50

3.9.1. undefined 50

3.9.2. object..... 50

3.9.3. boolean..... 51

3.9.4. number 51

3.9.5. string..... 51

3.9.6. function 51

3.9.7. bigint..... 52

3.9.8. symbol 52

3.10. Explorando números..... 52

3.10.1. Operadores aritméticos 53

3.10.2. Operadores de asignación 53

3.10.3. Métodos útiles 54

3.10.4. El objeto Math..... 55

3.10.5. Otros números 56

3.11. Explorando el objeto Dates 56

3.12. Declaraciones condicionales..... 59

3.12.1. Operadores de comparación matemática 59

3.12.2. Operadores de igualdad..... 59

3.12.3. Operadores lógicos..... 60

3.12.4. El operador NOT (!) 60

3.12.5. Igualdad en JavaScript 61

3.12.6. El operador de coalescencia nula (??)..... 63

3.12.7. La declaración if 63

3.12.8. La declaración switch... 65

3.12.9. Operador ternario 66

3.13. Entendiendo los bucles..... 68

3.13.1. while 68

3.13.2. do...while 68

3.13.3. for..... 69

3.14. Uso de cadenas de texto en JavaScript 70

3.14.1. Métodos importantes... 71

3.15. Explorando *arrays* 73

3.15.1. Operaciones básicas 74

3.15.2. Iterando sobre un *array* 77

3.15.3. Filtrado 79

3.15.4. Desestructuración 82

3.15.5. Set..... 83

3.16. Uso de objetos en JavaScript..... 84

3.16.1. Operaciones básicas 84

3.16.2. Iteración 87

3.16.3. Shallow copy versus deep copy	89
3.16.4. Fusionando objetos.....	92
3.16.5. Desestructuración	93
3.16.6. Encadenamiento opcional (?)	93
3.17. Explorando las funciones ...	95
3.17.1. Los fundamentos	95
3.17.2. Funciones flecha (<i>arrow functions</i>)	97
3.17.3. Closures	100
3.18. Creando y administrando clases	102
3.18.1. Creando una clase	103
3.18.2. Métodos de clase	103
3.18.3. Extendiendo clases	104
3.18.4. Métodos estáticos.....	105
3.18.5. Getters y setters	106
3.19. Resumen	107
3.19.1. Lectura adicional.....	108

CAPÍTULO 4

Programación asíncrona	109
4.1. Requisitos técnicos	110
4.2. Programación asíncrona en JavaScript	110
4.2.1. La mentalidad asíncrona.....	111
4.2.2. Métodos asíncronos en JavaScript.....	112
4.3. Entendiendo los <i>callbacks</i>	113
4.3.1. Temporizadores e intervalos.....	115
4.3.2. Error first en <i>callbacks</i> ...	118
4.3.3. Infierno de <i>callbacks</i>	120
4.4. Dominando las promesas.....	121
4.4.1. Uso de promesas.....	122
4.4.2. Creando promesas	124
4.4.3. Infierno de los <i>callbacks</i> con promesas	125

4.4.4. Promesas en paralelo.....	127
4.4.5. Manejo de errores.....	130
4.4.6. Encadenamiento de promesas.....	132
4.5. Uso de <i>async</i> y <i>await</i> para manejar código asíncrono	133
4.5.1. <i>Async</i>	133
4.5.2. <i>Await</i>	134
4.6. Funciones autoejecutables (IIFE)	136
4.7. Resumen	137
4.7.1. Lectura adicional	137

PARTE 2

ECOSISTEMA Y

ARQUITECTURA DE NODE.JS	139
--------------------------------------	------------

CAPÍTULO 5

Librerías del core de Node.js	141
5.1. Requisitos técnicos	142
5.2. Modularizando su código (ESM versus CJS).....	142
5.2.1. CommonJS (CJS)	144
5.2.2. ECMAScript Modules (ESM)	148
5.2.3. Entendiendo cómo funciona la interoperabilidad	151
5.3. Funcionamiento de las <i>core libraries</i>	154
5.3.1. La estructura de la librería.....	154
5.3.2. Índice de estabilidad	158
5.3.3. Otras librerías del <i>core</i> ...	159
5.3.4. Command-line options....	161
5.4. Resumen	163
5.4.1. Lectura adicional	164

CAPÍTULO 6

Librerías de terceros y npm	165
6.1. Requisitos técnicos	166

6.2. Administrando su aplicación con el <i>package.json</i>	166
6.3. Elegir las dependencias correctas para su proyecto.....	169
6.3.1. Riesgos.....	170
6.3.2. Buenos criterios.....	172
6.4. Instalación de dependencias.....	177
6.4.1. Local o global.....	177
6.4.2. Dependencia o dependencia de desarrollo.....	178
6.4.3. Añadiendo una nueva dependencia.....	179
6.4.4. Instalando todas las dependencias.....	182
6.4.5. Eliminando dependencias.....	182
6.5. Entendiendo el <i>package-lock.json</i>	183
6.6. Gestionando versiones de las dependencias.....	185
6.6.1. Dependencias desactualizadas.....	185
6.7. Construyendo JavaScript isomórfico.....	186
6.8. Usando <i>scripts</i> de <i>npm</i>	187
6.9. Ejecutando paquetes directamente con <i>npx</i>	189
6.10. Alternativas a <i>npm</i>	192
6.10.1. Yarn.....	192
6.10.2. PNPM.....	193
6.10.3. Verdaccio.....	193
6.11. Publicando nuestro primer paquete.....	193
6.11.1. Registros.....	194
6.11.2. Cuenta de <i>npm</i>	194
6.11.3. Preparar el paquete.....	194
6.11.4. Revisando el paquete..	197
6.11.5. Publicar el paquete.....	199
6.11.6. Nombres de paquetes.....	200
6.11.7. Lanzar nuevas versiones.....	201
6.11.8. Prevenir la publicación accidental.....	203
6.11.9. Mejores prácticas.....	203
6.12. Resumen.....	204
6.12.1. Lectura adicional.....	205
CAPÍTULO 7	
Arquitectura orientada a eventos.....	
eventos.....	207
7.1. Requisitos técnicos.....	208
7.2. Introducción a los eventos.....	208
7.3. Observando cambios en los archivos.....	209
7.4. La librería emisora de eventos de <i>Node.js</i>	211
7.4.1. Previendo el caos organizando los suscriptores.....	213
7.4.2. Eliminando suscriptores cuando no son necesarios.....	214
7.5. Su primer servidor HTTP.....	214
7.6. Agregando una interfaz de eventos a sus módulos.....	216
7.7. Resumen.....	218
7.7.1. Lectura adicional.....	219
CAPÍTULO 8	
Testíng en <i>Node.js</i>.....	
Testíng en <i>Node.js</i>.....	221
8.1. Requisitos técnicos.....	222
8.2. ¿Por qué es importante el <i>testíng</i> ?.....	222
8.2.1. El <i>testíng</i> es una cultura compleja.....	223
8.2.2. El <i>testíng</i> es una actividad de equipo.....	223
8.2.3. Aproveche ahora.....	224

8.3. Enfoque y filosofía del <i>testing</i>	226
8.3.1. El <i>testing</i> es ampliamente utilizado en el mundo	226
8.3.2. La pirámide de test	228
8.4. ¿Cómo debo probar mi código?	230
8.4.1. <i>Frameworks</i> y librerías ...	230
8.4.2. Nuestro primer test	231
8.4.3. Principios y objetivos del <i>testing</i>	232
8.5. Escribiendo nuestro primer <i>test suite</i>	234
8.5.1. Módulo <i>Utils</i>	234
8.5.2. La librería <i>node:test</i>	235
8.5.3. Usando la librería <i>Jest</i>	238
8.6. Dominando las herramientas de <i>test coverage</i>	242
8.6.1. Configuración	242
8.6.2. Ejecutando los test	243
8.6.3. Informe de <i>Node.js</i>	244
8.6.4. Informe de <i>Jest</i>	245
8.6.5. Informes interactivos	246
8.7. TDD en acción	247
8.8. Resumen	251
8.8.1. Lectura adicional	252

PARTE 3

FUNDAMENTOS DE UNA

APLICACIÓN WEB

CAPÍTULO 9

Manejo de HTTP y REST API

9.1. Requisitos técnicos	256
9.2. Cómo funciona Internet por debajo	256
9.2.1. Historia de Internet	257
9.2.2. Infraestructura de Internet	258

9.2.3. Request for Comments (RFC)	259
9.2.4. Otros RFC	262
9.3. HTTP – Relación servidor y cliente	263
9.3.1. Petición y respuesta	263
9.3.2. Renderizado del lado del servidor (<i>server render</i>)	266
9.3.3. Single Page Applications (SPA)	267
9.4. Dominando HTTP	269
9.4.1. Encabezados HTTP	269
9.4.2. Códigos de estado	272
9.4.3. Métodos de petición	275
9.4.4. Cargas útiles de HTTP	277
9.4.5. Versiones de HTTP	277
9.5. Uso de URL en aplicaciones web	278
9.6. REST API en resumen	280
9.7. Explorando la especificación <i>JSON</i>	284
9.8. Depuración de peticiones HTTP	285
9.9. Resumen	286
9.9.1. Lectura adicional	287

CAPÍTULO 10

Desarrollando aplicaciones

web con *Express*

10.1. Requisitos técnicos	290
10.2. Familiarizándonos con la librería <i>Express</i>	290
10.2.1. Instalando <i>Express</i>	291
10.2.2. Hola Mundo	291
10.2.3. Usando el generador	293
10.2.4. Depuración	295
10.3. Entendiendo los motores de plantillas	297
10.3.1. Elegir un motor de plantillas	297

10.3.2. Renderizando una plantilla.....	297
10.3.3. Entendiendo el proceso .	300
10.4. Dominando las peticiones..	301
10.4.1. Métodos HTTP	302
10.4.2. Rutas.....	303
10.4.3. Controladores	307
10.5. Dominando las respuestas.....	309
10.5.1. Gestión de cabeceras ..	309
10.5.2. Códigos de estado.....	311
10.5.3. Redirecciones.....	312
10.5.4. Enviando datos.....	313
10.5.5. Enviando archivos	315
10.6. Usando el patrón de <i>middleware</i>	316
10.6.1. Entendiendo el alcance	320
10.6.2. Anatomía del <i>middleware</i>	320
10.6.3. Agregando <i>middleware</i> a la aplicación	323
10.6.4. Agregando <i>middleware</i> a una ruta	324
10.6.5. <i>Middleware</i> común utilizado en <i>Express</i>	325
10.7. Uso de <i>middleware</i> de terceros	327
10.8. Resumen	328
10.8.1. Lectura adicional.....	329

PARTE 4

CONSTRUYENDO

APLICACIONES WEB SÓLIDAS

CON NODE.JS 331

CAPÍTULO 11

Construyendo una aplicación

web desde cero 333

11.1. Requisitos técnicos..... 334

11.2. El inicio del proyecto334

11.2.1. El objetivo del proyecto334

11.2.2. Comenzando con el primer paso336

11.3. Construyendo una REST API337

11.3.1. Añadiendo las rutas.....338

11.3.2. Añadiendo los *stores*....340

11.3.3. Añadiendo los archivos estáticos 349 |

11.3.4. Añadiendo las plantillas 350 |

11.4. *Testing* con *supertest*.....351

11.4.1. Añadiendo *stores* al servidor 352 |

11.4.2. Creando utilidades de test 354 |

11.4.3. Añadiendo test al servidor 357 |

11.5. Revisando el resultado final del proyecto.....367

11.5.1. La página Acerca de.....367

11.5.2. ¡Celebremos!.....371

11.5.3. Próximos pasos.....372

11.6. Resumen.....372

CAPÍTULO 12

Persistencia de datos

con MongoDB 373

12.1. Requisitos técnicos374

12.2. ¿Qué es MongoDB?.....374

12.2.1. Versiones.....375

12.3. Configuración de MongoDB 376 |

12.3.1. Instalación de Docker...376

12.3.2. Ejecutando MongoDB en un contenedor.....377

12.3.3. Ejecutando MongoDB con Docker Compose.....379

12.3.4. Agregando comandos	13.2.1. Autenticación	413
Docker a package.json.....	13.2.2. Autorización	413
381	13.3. JWT en pocas palabras	415
12.3.5. Conectándose a	13.3.1. El proceso.....	416
MongoDB.....	13.3.2. La teoría	417
382	13.3.3. Criptografía 101	417
12.3.6. Otras formas de	13.3.4. Estructura de JWT.....	420
instalar MongoDB.....	13.3.5. JWT.io	421
383	13.4. Entendiendo los	
12.4. Cómo usar secretos en	fundamentos de <i>Passport.js</i>	422
Node.js	13.5. Agregando autenticación	
383	y autorización a nuestra	
12.4.1. Variables de entorno....	aplicación web.....	423
384	13.5.1. Descargando el	
12.4.2. El archivo .env	proyecto base.....	423
385	13.5.2. Configuración.....	423
12.4.3. dotenv	13.5.3. Ejecute los test	424
385	13.5.4. Los cambios en la UI	426
12.5. Usando un ODM -	13.5.5. Dependencias	
Mongoose.....	añadidas.....	430
386	13.5.6. Cambios en el	
12.6. Migrando una aplicación	<i>frontend</i>	431
web a MongoDB.....	13.5.7. Se añadió un nuevo	
387	<i>store</i> para usuarios.....	433
12.6.1. Instalando	13.5.8. Utilidades <i>JWT</i>	437
dependencias.....	13.5.9. Añadiendo nuevas	
388	rutas	439
12.6.2 Administrando secretos ..	13.5.10. Mejoras en las	
388	utilidades de testing	442
12.6.3. Migrando la capa de	13.5.11. Cambios en los test....	442
datos	13.5.12. Cobertura de test	444
391	13.6. Resumen.....	445
12.7. Probando nuestra capa	13.6.1. Lectura adicional	445
de integración con MongoDB.....		
396		
12.7.1. Actualizar las		
utilidades.....		
396		
12.7.2. Refactorizando el test		
<i>suite</i>		
398		
12.7.3. Algunos test deben		
cambiar		
400		
12.7.4. Verificando la		
cobertura.....		
402		
12.8. Resumen		
407		
12.8.1. Lectura adicional.....		
408		

CAPÍTULO 13

Autenticación y autorización

de usuarios 411

13.1. Requisitos técnicos..... 412

13.2. Entendiendo la autenticación y la autorización moderna 412

CAPÍTULO 14

Manejo de errores en Node.js.... 447

14.1. Requisitos técnicos

448

14.2. Explorando los tipos de errores

448

14.2.1. Errores de sintaxis.....

449

14.2.2. Errores en tiempo de ejecución	449
14.3. Lanzando errores claros.....	450
14.3.1. El objeto de error.....	451
14.3.2. Errores personalizados.....	452
14.4. Manejo de errores en Express	454
14.4.1. <i>Middleware</i> para el manejo de errores	454
14.4.2. Errores personalizados.....	455
14.5. Graceful shutdown en aplicaciones.....	456
14.5.1. Eventos	457
14.5.2. Códigos de salida	459
14.5.3. Evitando procesos zombi	460
14.6. Resumen	461
14.6.1. Lectura adicional.....	462

CAPÍTULO 15

Proteger aplicaciones web.....	463
15.1. La importancia de la seguridad.....	464
15.1.1. Ingeniería social	465
15.1.2. Riesgos en la cadena de suministro	467
15.1.3. El impacto en el negocio	468
15.2. Cómo comenzar con la seguridad.....	469
15.2.1. Resumen del OWASP Top 10	469
15.2.2. Common Weakness Enumeration (CWE)	471
15.2.3. Common Vulnerabilities and Exposures (CVE)	472

15.2.4. Modelo de amenazas de Node.js.....	476
15.2.5. Las recomendaciones oficiales de Node.js.....	476
15.3. Mejorando la seguridad de nuestras aplicaciones.....	477
15.3.1. Póngalo en práctica.....	481
15.4. Convirtiéndose en un <i>hacker</i> ético	482
15.4.1. Divulgación coordinada de vulnerabilidades (CVD)	483
15.4.2. Programas de recompensas.....	485
15.4.3. Adquiriendo las habilidades.....	486
15.5. Resumen.....	486
15.5.1. Lectura adicional	487

PARTE 5

DOMINANDO EL DESPLIEGUE

Y LA PORTABILIDAD DE

NODE.JS	489
----------------------	------------

CAPÍTULO 16

Desplegando aplicaciones

de Node.js	491
16.1. Requisitos técnicos	492
16.2. Definiendo los requisitos	492
16.3. Usando un repositorio de GitHub	496
16.3.1. Creando un repositorio de GitHub.....	496
16.3.2. Subiendo el código al repositorio.....	497
16.4. Integración continua con GitHub Actions	499
16.5. Usando MongoDB Atlas	504
16.6. Despliegue de aplicaciones Node.js con PM2	506

16.6.1. Creando un <i>droplet</i> de DigitalOcean	507
16.6.2. Conectándose a la máquina	508
16.6.3. Preparando la máquina	509
16.6.4. Clonando el repositorio	510
16.6.5. Instalando las dependencias	511
16.6.6. Preparando el entorno	511
16.6.7. Administrando la aplicación con PM2	512
16.6.8. Accediendo a la aplicación	512
16.7. Resumen	514
16.7.1. Lectura adicional	514

CAPÍTULO 17

Dockerizando una aplicación

de Node.js	515
17.1. Requisitos técnicos.....	516
17.2. Contenedores y soluciones nativas de la nube con Docker	517
17.2.1. Ciclo de vida de Docker	517
17.2.2. <i>Dockerizando</i> la aplicación	518
17.2.3. Administrando la aplicación con Docker	520
17.2.4. Agregando la construcción de Docker a la CI	522
17.2.5. Subiendo la imagen a Docker Hub	523
17.2.6. Publicando la imagen con GitHub Actions	525

17.3. Ejecutando los contenedores.....	526
17.4. Usando Cloudflare.....	529
17.5. Siguiendo nivel - Principios de la aplicación Twelve-Factor.....	531
17.6. Limpieza.....	532
17.7. Resumen.....	533
17.7.1. Lectura adicional	534