

Podrá descargar algunos elementos de este libro en la página web de Ediciones ENI: **<http://www.ediciones-eni.com>**.
Escriba la referencia ENI del libro **RIT2TYP** en la zona de búsqueda y valide. Haga clic en el título y después en el botón de descarga.

Prefacio

Capítulo 1

Introducción

1. Un poco de historia	9
2. ECMAScript	13
3. ¿Por qué TypeScript?	17
3.1 Transpilación	17
3.2 Tipado estático	20
4. Los entresijos del tipado	22
4.1 Archivos de definición	22
4.2 Tipado estructural	24
5. La arquitectura de TypeScript	27
5.1 Core TypeScript Compiler	27
5.2 Standalone TS Compiler	28
5.3 Language Service	28
5.4 Opciones de compilación	28
6. Entorno de desarrollo	30
6.1 Node.js	30
6.2 Visual Studio Code	31
6.3 Instalar TypeScript	39

Capítulo 2

Tipos e instrucciones básicas

1. Variable y alcance	43
2. Tipos básicos	47
2.1 Introducción	47
2.2 Tipos primitivos básicos	47
2.3 Tipos primitivos: undefined y null	51
2.4 Tipos por referencia: arreglos y tuplas	54
2.5 Tipos por referencia: object	57
2.6 Tipos especiales: any y void	60
2.7 Aserción de tipo	62
2.8 Tipo especial: unknown	63
3. Desestructuración	66
4. Enumeración	70
5. Condicional If...Else	77
6. Optional chaining	79
7. Operadores	80
7.1 Operadores aritméticos	81
7.2 Operadores unarios	81
7.3 Operadores de comparación	82
7.4 Operadores lógicos combinatorios	83
8. Zoom en los operadores	83
8.1 Operadores unarios + y -	84
8.2 Operador unario !	84
8.3 Operadores == y ===	85
8.4 Operadores y &&	86
8.5 Operador de coalescencia nula (??)	88
8.6 Operador de propagación	88
9. Bucles	90
9.1 Bucles for...in	90
9.2 Bucle for...of	92

10. Symbol	93
11. Iteración y colecciones	97
11.1 Iterador	97
11.2 Generadores.	99
11.3 Map y set.	100
11.3.1Map	100
11.3.2Set	101
11.3.3WeakMap y WeakSet	102
12. Funciones	102
12.1 Los conceptos básicos.	102
12.2 Parámetro rest.	104
12.3 Closures.	107
12.4 This y las funciones flecha.	108
12.5 Tipo función	113
13. Prototipo	114
14. Gestión de excepciones.	115

Capítulo 3

Programación orientada a objetos

1. Introducción	119
2. Las clases	120
3. Propiedades	123
4. Métodos.	126
5. Constructores	132
6. Métodos estáticos.	134
7. Modificadores de acceso.	137
8. Encapsulación	145
9. Herencia	148
10. Abstracción	155
11. Interfaces.	159

12. Polimorfismo	170
13. Los principios SOLID	172
13.1 Introducción a los principios SOLID	172
13.2 El principio de responsabilidad única	172
13.3 El principio de abierto a la extensión, cerrado a la modificación	175
13.4 El principio de sustitución de Liskov	178
13.5 El principio de segregación de interfaces	183
13.6 El principio de inversión de dependencias	185

Capítulo 4

Módulos

1. Introducción	189
2. Histórico	190
2.1 Pattern module	190
2.2 AMD y CommonJs	192
2.3 Estandarización	197
3. El estándar ECMAScript 2015	197
3.1 Export	201
3.2 Import	205
3.3 Import assert	208
4. Gestión de tipos	208
5. Carga y resoluciones	210
5.1 Resoluciones	210
5.2 Carga	213

Capítulo 5

Genericidad

1. Introducción	215
2. Declaración básica	215
3. Clases e interfaces	218

4. Restricciones	221
5. Genérico y constructor	227

Capítulo 6 Decoradores

1. Introducción	229
2. Decoradores experimentales	231
2.1 Decoradores experimentales de clase	232
2.2 Decoradores de métodos experimentales	234
2.3 Decoradores experimentales de propiedad	237
2.4 Decoradores experimentales de parámetro	241
2.5 Fábrica de decoradores	244
2.6 Metadatos	246
3. Decoradores ECMAScript	255
3.1 Decoradores ECMAScript de clase	257
3.2 Decoradores ECMAScript de método	259
3.3 Decoradores ECMAScript de propiedad	261
3.4 Decoradores ECMAScript de accesores	264
3.4.1 Decorador "getter"	264
3.4.2 Decorador "setter"	266
3.4.3 Decorador de accesores automáticos	267

Capítulo 7 Asincronismo

1. Introducción	273
2. Callback hell	274
3. Promesa	276
3.1 Histórico	276
3.2 Promesas de ECMAScript 2015	277
3.2.1 Resolución de promesas	277
3.2.2 Then, catch y finally	279
3.2.3 Métodos estáticos	284

3.2.4 Conclusión	285
4. Async/await	286

Capítulo 8

Sistema de tipo avanzado

1. Introducción	291
2. Alias de tipo	291
3. Tipo never	295
4. Tipo unión e intersección	297
4.1 Unión	297
4.2 Intersección	299
5. Type guards	302
5.1 Introducción	302
5.2 Operador typeof	303
5.3 Operador instanceof	304
5.4 Operador in	306
5.5 Unión discriminante	309
5.6 Type guards definidas por el usuario	311
5.7 Gestión de null y undefined	314
5.8 Caso imposible	317
5.9 Función de afirmación	318
6. Tipos literales	320
7. Modelo de tipo literal	322
8. Tipo index	326
9. Mapped type	331
9.1 Antes de TypeScript 2.1	331
9.2 Operadores	334
9.3 Remapeado de claves	335
9.4 Mapped type nativo	336
10. Afirmación constante	339
11. Tuplas variádicas	341

12. Tipo condicional	343
12.1 Conceptos básicos.	343
12.2 Distributividad	345
12.3 Operador infer.	348
12.4 Tipo condicional nativo	349
13. Operador satisfies	353

Capítulo 9

TypeScript y programación funcional

1. Introducción	357
2. Función pura e impura	359
3. Inmutabilidad	363
4. Iteración	370
5. Condiciones.	374
6. Función parcial	375
7. Currying	376
8. Pattern Matching	379
9. Composición de funciones.	382
10. Para ir más lejos...	386

Capítulo 10

Un primer proyecto con Node.js

1. Introducción	387
2. Configurar el proyecto	388
2.1 Crear el archivo package.json.	389
2.2 Configurar TypeScript	390
2.2.1 Instalación.	390
2.2.2 Configuración de TypeScript.	391
2.3 Una primera aplicación	398
3. Crear el framework MVC	401
3.1 Registro de rutas.	402

3.2	Fábrica de controladores.	405
3.3	El decorador Controller	407
3.4	La fábrica de decoradores Action	408
3.5	Crear el controlador EmployeeController.	410
3.6	Inicializar la aplicación.	412
4.	Validación y OpenApi	417
4.1	Tipado de esquemas	418
4.2	El decorador de propiedad SchemaProperty	423
4.3	Vínculo entre una acción y un modelo	424
4.4	El decorador de método Model	427
4.5	Administrar esquemas en la clase Server	430
4.6	El explorador de API.	431
4.7	La acción post del controlador EmployeeController	432
5.	Acceso a datos y su persistencia	437
5.1	Tipado de las entidades de trabajo	438
5.2	Persistencia de datos.	439
5.3	Repository.	441
6.	Inversión de dependencias	446
6.1	El contenedor de dependencias	447
6.2	El decorador Inject	450
6.3	Inyección de dependencias.	451
6.4	Almacenamiento de datos en un archivo	453
7.	Para ir más lejos	455
	Índice	457

Podrá descargar algunos elementos de este libro en la página web de Ediciones ENI: **<http://www.ediciones-eni.com>**.
 Escriba la referencia ENI del libro **EITREACT** en la zona de búsqueda y valide. Haga clic en el título y después en el botón de descarga.

Capítulo 1 Prólogo

1. Introducción	17
1.1 A quién va dirigido este libro	17
1.2 ¿Por qué elegir React?	18
1.2.1 Reactividad	18
1.2.2 Composabilidad	18
1.2.3 Comunidad activa	18
1.2.4 React Native	19
1.3 Cómo utilizar este libro	19
1.4 Acerca de los ejemplos de código	20
2. ¿Qué es React?	20
2.1 Introducción a React	20
2.1.1 Origen de React	20
2.1.2 Objetivo de React	20
2.1.3 ¿Por qué es tan popular React?	20
2.1.4 Principales conceptos de React	21
2.2 Filosofía React	22
2.2.1 Componentes: la unidad básica	22
2.2.2 Reactividad: actualización eficaz de la interfaz de usuario	23
2.2.3 Composición: ensamblaje de componentes reutilizables	23
2.2.4 Un enfoque declarativo	23
2.3 La comunidad React	24
2.3.1 Proyectos de código abierto	24
2.3.2 Contribuidores comprometidos	25
2.3.3 Eventos y conferencias	25

2 **React**

Desarrolle el Front End de sus aplicaciones web y móviles

2.4	Versiones de React y ciclo de actualización	25
2.4.1	Las diferentes versiones de React	25
2.4.2	Principales cambios y mejoras	26
2.4.3	Seguimiento de nuevas versiones	26
3.	Estructura del libro	27
3.1	Descubrimiento de React	27
3.2	Gestión de estados en React	27
3.3	Desarrollo móvil con React Native	27
3.4	Profundizar en React	28

Capítulo 2

Empezar bien con React

1.	Conceptos básicos de JavaScript	29
1.1	El DOM (Document Object Model)	30
1.2	Definición de una variable en JavaScript	31
1.3	Los tipos de valores	32
1.4	Cadenas de caracteres	34
1.5	Arrays	35
1.6	Objetos	36
1.7	Clases	38
1.8	Funciones flecha	40
1.8.1	Sintaxis de las funciones flecha	40
1.8.2	Comparación entre una función tradicional y una función flecha	41
1.8.3	Uso de funciones flecha en React	41
1.9	Asincronismo	42
1.9.1	Callbacks	42
1.9.2	Promesas (Promises)	43
1.9.3	Async/Await	44

2.	Prerequisitos, conceptos y herramientas para desarrollar	45
2.1	Instalación de las herramientas necesarias	45
2.1.1	NodeJS.	45
2.1.2	Editor.	46
2.1.3	Navegador web y herramientas de desarrollo.	46
3.	Una aplicación básica	47
3.1	El mínimo de código para una aplicación React	47
3.2	Los componentes de React.	49
4.	Entorno de desarrollo	50
4.1	Introducción a JSX	50
4.2	Inicialización de un proyecto con Vite	51
4.3	Configuración de Prettier en VSCode	52
4.4	Configuración de ESLint	53
4.5	Conclusión	55

Capítulo 3

Descubrir el JSX

1.	Introducción a JSX	57
2.	Sintaxis y elementos JSX	60
2.1	Elementos JSX.	60
2.2	Atributos JSX	61
2.2.1	Atributos booleanos	61
2.2.2	Atributos personalizados	62
2.3	Elementos y fragmentos de React	63
2.4	Comentarios JSX	64
2.5	Uso de JavaScript dentro de JSX	65
3.	Construir una interfaz con componentes	65
3.1	Composición.	66
3.2	Aislamiento de funcionalidades.	67
3.3	Profundidad de la composición	70

3.4	Buenas prácticas	71
3.4.1	División en componentes reutilizables	72
3.4.2	Profundidad de composición razonable	72
3.4.3	Respeto al principio de responsabilidad unica	72
4.	Los fragmentos JSX	72
4.1	Utilización de fragmentos JSX	73
4.2	Fragmento explícito	74
5.	Expresiones condicionales	74
5.1	El operador ternario	75
5.2	La sentencia if	75
6.	Utilizar listas y claves.	76
6.1	Las claves de los elementos de la lista	76
6.1.1	¿Por qué son importantes las claves?	76
6.1.2	¿Cómo asignar claves?	77
6.2	Utilizar una lista.	77
6.3	Las claves y el rendimiento	79

Capítulo 4

Los fundamentos de React

1.	Los componentes	81
1.1	Introducción a los componentes	81
1.1.1	Tipos de componentes	81
1.1.2	Crear un componente.	82
1.1.3	Importar un componente.	83
1.2	Utilización de props	84
1.2.1	Pasar props a un componente	84
1.2.2	Uso de props en el componente hijo	85
1.3	Diferencias entre componentes funcionales y de clase	86
1.4	Ciclo de vida de los componentes de clase	87

1.5	Componentes controlados y no controlados	88
1.5.1	Componentes controlados	88
1.5.2	Componentes no controlados	89
1.5.3	¿Cuándo debe utilizarse cada enfoque?	90
1.6	Componentes con estado (stateful) y sin estado (stateless)	90
1.6.1	Componentes con estado	91
1.6.2	Componentes sin estado	91
1.6.3	¿Cuándo debe utilizarse cada tipo de componente?	92
1.7	Composición y jerarquía de los componentes	92
2.	Los props y el estado local	93
2.1	Introducción a los props y al estado local	93
2.1.1	Props	94
2.1.2	El estado (state)	95
2.2	Utilización de props	97
2.2.1	Transmisión de datos de padres a hijos	97
2.2.2	Utilizar props por defecto	98
2.3	Estado local	99
2.3.1	Utilización del hook useState para gestionar el estado local	99
2.3.2	Modificar el estado local con setState (para componentes de clase)	100
2.3.3	Comparación entre props y el estado local	102
3.	Los hooks	103
3.1	Introducción a los hooks	103
3.2	Problemas con los componentes de clase	103
3.3	Los principales hooks incorporados	104
3.3.1	useState	104
3.3.2	useEffect	105
3.3.3	useContext	107
3.3.4	useReducer	109
3.3.5	useCallback	110
3.3.6	useMemo	112
3.3.7	useRef	114

6 **React**

Desarrolle el Front End de sus aplicaciones web y móviles

3.3.8	useImperativeHandle	115
3.3.9	useLayoutEffect	117
3.3.10	useDebugValue	118
3.4	Crear sus propios hooks	120
3.5	Buenas prácticas para el uso de hooks	121
3.5.1	Respete el orden de los hooks y no los llame condicionalmente	123
3.5.2	Divida las responsabilidades de los hooks para evitar complejidades	124
3.6	Limitaciones de los hooks	126
4.	El estilo	126
4.1	Introducción al estilo en React	126
4.1.1	Utilización de estilos en línea (Inline Styles)	126
4.1.2	Utilización de clases CSS	127
4.1.3	Utilización de módulos CSS	128
4.1.4	Uso de bibliotecas de estilo (por ejemplo, Bootstrap, Material-UI)	130
4.2	Tailwind	131
4.3	La gestión de clases condicionales	132
4.4	Animaciones y transiciones	136
5.	Enrutamiento	138
5.1	Introducción al enrutamiento en React	138
5.1.1	Comprender el concepto de enrutamiento en una aplicación web	138
5.1.2	React Router	139
5.2	Utilización de parámetros URL (URL Parameters)	140
6.	Formularios	141
6.1	Utilización de formularios controlados	141
6.1.1	Gestión del estado del formulario con useState	142
6.1.2	Validación de campos del formulario	143
6.1.3	Envío del formulario	144

6.2	Uso de formularios no controlados	146
6.2.1	Definición de formas no controladas	146
6.2.2	Gestión de datos de formularios con referencias (refs)	147
6.2.3	Acceso a los valores de los formularios enviados	148
6.3	Utilización de componentes de formulario.	149
6.3.1	Presentación de Formik	149
6.3.2	Instalación y configuración de Formik	150
6.3.3	Creación de un formulario con Formik.	150
6.3.4	Validación de campos con Formik.	151
6.4	Campos de formulario avanzados	153
6.4.1	Seleccionar campo	153
6.4.2	Casillas de verificación (checkbox) y botones de opción (radio button)	154
6.4.3	Campo de archivo (file input)	156
6.4.4	Selector de fecha (date picker)	157
6.5	La validación de formularios	159
6.6	React-hook-form, la alternativa a Formik.	161
6.7	Gestión de formularios complejos	162
6.7.1	Formularios con campos dependientes	162
6.7.2	Formularios Multipasos	163
6.7.3	Formularios con listas dinámicas (añadir/eliminar campos).	166
7.	Optimización del rendimiento de las aplicaciones.	168
7.1	Comprender los problemas de rendimiento	168
7.1.1	Reactividad de la interfaz de usuario	169
7.1.2	Velocidad de carga inicial	169
7.1.3	Gestión de estado	169
7.1.4	Renderizado no necesario.	170
7.1.5	Consulta y gestión de datos.	170
7.1.6	Renderizado condicional	170
7.2	Almacenamiento en caché y memorización	170
7.3	Lazy loading y suspense	171
7.4	Virtualización	172

7.5	Code splitting	173
7.6	Análisis de rendimiento	173
7.7	Prueba de rendimiento	173

Capítulo 5

Gestión del estado

1.	Introducción	175
2.	Las diferentes soluciones	175
2.1	Estado del servidor (server state) y estado del cliente (client state)	176
2.2	El context API	176
2.3	Redux	176
2.4	MobX	177
2.5	Recoil	177
2.6	Zustand	177
3.	Combinar contexto y los hooks	178
3.1	Crear un contexto	178
3.2	Utilización de Context con el hook useContext	179
3.3	Uso de useReducer para estados más complejos	180
4.	Redux	182
4.1	Elementos clave de Redux	182
4.1.1	Actions	182
4.1.2	Reductor (reducer)	183
4.1.3	Store	183
4.1.4	Redux Toolkit	184
4.2	Crear un store	184
4.3	Utilizar createSlice	185
4.4	Asincronismo con createAsyncThunk	186
4.5	Capacidad de prueba	187

5. MobX.....	188
5.1 Principios básicos de MobX.....	188
5.1.1 Observables.....	188
5.1.2 Actions.....	189
5.1.3 Reactions.....	189
5.1.4 Computed values.....	189
5.2 MobX con React.....	190
5.3 Conclusión.....	191

Capítulo 6

Gestión del estado del servidor con React Query

1. Introducción.....	193
2. Instalación.....	193
2.1 Instalación de dependencias.....	193
2.2 Configuración del proveedor.....	194
3. Consultas (queries).....	195
3.1 La función fetch.....	195
3.2 Utilización de useQuery.....	196
4. Mutaciones.....	198
5. Invalidation.....	200
5.1 Diferentes formas de invalidar una solicitud.....	200
5.2 Otros medios de invalidación.....	201
5.3 Invalidaciones condicionales.....	201
6. Utilizar herramientas de desarrollo específicas.....	202

Capítulo 7**Desarrollo móviles con React Native**

1. Introducción	205
2. Presentación de React Native	205
2.1 Componentes nativos	205
2.2 Reutilizar la lógica	206
2.3 Acceso a las funciones del dispositivo	206
3. Crear una aplicación con Expo	206
3.1 La herramienta create-expo-app	207
3.2 La aplicación Expo Go	207
4. Configurar la navegación	209
4.1 Instalación de dependencias y configuración del proyecto	209
4.2 El directorio app	211
4.3 Navegación entre pantallas	212
5. Añadir estilo	213
5.1 Utilizar estilos en línea	213
5.2 Utilización de estilos externos	214
5.3 Estilos condicionales	215
5.4 Utilizar Tailwind	216

Capítulo 8**Funciones avanzadas con React Native**

1. Introducción	219
2. Utilizar la geolocalización y acceder a los contactos	220
2.1 Utilizar la geolocalización	220
2.1.1 Instalación del módulo de geolocalización	220
2.1.2 Utilización de la geolocalización	220
2.2 Acceso a los contactos	222
2.2.1 Instalación del módulo de contacto	222
2.2.2 Utilizar el acceso a los contactos	223

3. Ejemplo: utilización de un tema claro u oscuro con la API de contexto	225
4. Conclusión	228

Capítulo 9

Next.js, el framework React por Vercel

1. Introducción	229
1.1 ¿Por qué Next.js?	229
1.1.1 Renderizado del lado del servidor (SSR) y generación de sitios estáticos (SSG)	229
1.1.2 Enrutamiento sencillo e intuitivo	230
1.1.3 Precarga de páginas	230
1.1.4 Optimización de imágenes	230
1.1.5 Compatibilidad con CSS-in-JS y SCSS	230
2. Creación de un sitio con Next.js	231
2.1 Creación de un proyecto Next.js	231
2.2 Creación de páginas	232
2.3 Iniciar la aplicación	233
3. Navegación entre las páginas	233
3.1 Utilización del componente Link	233
3.2 Ruta dinámica	234
3.3 Navegación con useRouter	235
3.4 Navegación programada	235
4. Gestión de assets	236
4.1 Archivos públicos	236
4.2 Carga de imágenes	237
5. Utilizar SWR, el equivalente de React Query	238
5.1 Instalación de SWR	238
5.2 Utilización de SWR	239
5.3 Opciones de configuración avanzadas	240

6.	Renderizado del lado del servidor y generación de sitios estáticos .	241
6.1	Renderizado del lado del servidor (SSR)	241
6.2	Generación de sitios estáticos (SSG)	242
6.3	Actualización automática (ISR)	243
7.	Presentación de React Server Components (RSC)	243

Capítulo 10

Introducción a Storybook

1.	Descubrimiento e instalación	247
1.1	¿Qué es un Storybook?	247
1.2	Instalación de Storybook	248
1.3	Crear historias (stories)	249
1.4	Lanzamiento de Storybook	249
1.5	Qué ofrece Storybook.	250
2.	Creación de un componente	250
2.1	Creación de un componente	250
2.2	Visualización y comprobación del componente.	252
2.3	La utilidad de crear componentes en Storybook.	252
3.	Utilizar stories	253
3.1	Visualizar y poner a prueba la story	253
3.2	Ventajas de crear stories.	253
4.	Utilización de add-ons	254
4.1	Instalación de add-ons	254
4.2	Utilización de add-ons	255
4.3	Lo que ofrece el uso de add-ons	255
5.	Conclusión	256

Capítulo 11

Conectar React a una API GraphQL

1. Introducción a GraphQL y primeras consultas.	257
1.1 ¿Qué es GraphQL?	257
1.2 Primeras consultas GraphQL con React	258
2. Creacion de una API con Hasura	262
3. Llamada a una API con React y Apollo Client	266
3.1 Configuración del cliente Apollo	266
3.2 Llamada a la API creada con Hasura	266
4. Conclusión	267

Capítulo 12

Dominar los design patterns de React

1. Introducción	269
2. Higher-Order Components	270
2.1 Funcionamiento de los Higher-Order Components.	270
2.2 Qué ofrecen los Higher-Order Components.	271
2.3 Casos prácticos de Higher-Order Components	271
2.4 Limitaciones y consideraciones	272
2.4.1 Los HOC en cascada.	272
2.4.2 Conflicto de props	272
3. Renderizar props.	274
3.1 Cómo funcionan los render props	274
3.2 Cualidades de los render props	275
3.3 Casos de uso de render props.	275
3.3.1 Control de los componentes interactivos	275
3.3.2 Gestión del estado o de datos.	276
3.4 Limitaciones y consideraciones	277

4.	Provider	278
4.1	Funcionamiento del Provider Pattern	278
4.2	Ventajas del Provider Pattern	278
4.3	Casos prácticos del Provider Pattern	279
4.4	Limitaciones y consideraciones	279
5.	Componentes container y de presentación.	280
5.1	Componentes container.	280
5.2	Componentes de presentación	281
5.3	Qué ofrecen el componente container y los de presentación	281
5.4	Casos prácticos de componentes container y de presentación	282
5.5	Limitaciones y consideraciones	282
6.	Hooks	282
6.1	Conceptos básicos del pattern Hooks	283
6.2	Hooks personalizados.	283
6.3	Extracción lógica.	284

Capítulo 13

Probar una aplicación React

1.	Uso de Jest y React Testing Library	287
1.1	Introducción a Jest	287
1.2	Instalación de Jest.	288
1.3	Escribir pruebas con Jest y React Testing Library	288
1.4	Ejecución de pruebas	289
1.5	Jest y React Testing Library.	289
1.6	Limitaciones y consideraciones	289
2.	Escribir su primera prueba	290
2.1	Configuración básica	290
2.2	Escribir su primera prueba	290
2.3	Explicación de la prueba.	291
2.4	Ejecución de la prueba	291
2.5	¿Por qué esta sencilla prueba?	292

2.6	Limitaciones y consideraciones	292
3.	Probar los componentes con Cypress	292
3.1	Configuración básica	293
3.2	Instalación de Cypress	293
3.3	Escribir pruebas con Cypress	293
3.4	Ejecución de pruebas	294
3.5	Qué ofrece Cypress	294
3.6	Limitaciones y consideraciones	294
3.7	Playwriqh, la alternativa a Cypress	295
4.	Conclusión	296
4.1	La importancia de las pruebas	296
4.2	Elección del método de prueba	296
4.3	Buenas prácticas de prueba	296
4.4	Integración y despliegue continuos	297
4.5	Evolución de las pruebas	297

Conclusión

1.	Ir más allá	299
1.1	Explorar áreas relacionadas	299
1.2	Contribuir a la comunidad	300
1.3	Crear proyectos personales	300
1.4	Participación en eventos y conferencias	300
1.5	Experimentar con nuevas tecnologías	300
2.	Despliegue con Netlify	301
2.1	Creación de un proyecto y compilación	301
2.2	Crear una cuenta Netlify	301
2.3	Configuración del despliegue	301
2.4	Despliegue	302

3.	Utilizar una biblioteca de componentes	303
3.1	¿Por qué utilizar una biblioteca de componentes?	303
3.2	Ejemplos de bibliotecas de componentes	303
3.3	Limitaciones	303
4.	Recursos de lectura complementaria.	304
4.1	Documentación oficial	304
4.2	Blogs	304
4.3	Vigilancia tecnológica.	305
	Índice.	307