

---

# Índice

## de contenidos

---

<b>Prólogo .....</b>	<b>18</b>
<b>Introducción .....</b>	<b>24</b>
Sobre la imagen de cubierta .....	26
<b>1. Código limpio .....</b>	<b>28</b>
Hágase el código .....	29
Código incorrecto.....	30
El coste total de un desastre .....	31
El gran cambio de diseño.....	31
Actitud .....	32
El enigma .....	33
¿El arte del código limpio? .....	33
Concepto de código limpio.....	34
Escuelas de pensamiento .....	39
Somos autores.....	40
La regla del Boy Scout.....	41
Precuela y principios .....	41
Conclusión .....	42
Bibliografía.....	42

<b>2. Nombres con sentido.....</b>	<b>44</b>
Introducción.....	45
Usar nombres que revelen las intenciones.....	45
Evitar la desinformación.....	47
Realizar distinciones con sentido.....	48
Usar nombres que se puedan pronunciar.....	49
Usar nombres que se puedan buscar.....	50
Evitar codificaciones.....	51
Notación húngara.....	51
Prefijos de miembros.....	51
Interfaces e implementaciones.....	52
Evitar asignaciones mentales.....	52
Nombres de clases.....	53
Nombres de métodos.....	53
No se exceda con el atractivo.....	53
Una palabra por concepto.....	54
No haga juegos de palabras.....	54
Usar nombres de dominios de soluciones.....	55
Usar nombres de dominios de problemas.....	55
Añadir contexto con sentido.....	55
No añadir contextos innecesarios.....	57
Conclusión.....	58
<b>3. Funciones.....</b>	<b>60</b>
Tamaño reducido.....	63
Bloques y sangrado.....	64
Hacer una cosa.....	64
Secciones en funciones.....	65
Un nivel de abstracción por función.....	66
Leer código de arriba a abajo: la regla descendente.....	66
Instrucciones Switch.....	67
Usar nombres descriptivos.....	68
Argumentos de funciones.....	69
Formas monádicas habituales.....	70
Argumentos de indicador.....	70
Funciones diádicas.....	71
Triadas.....	71
Objeto de argumento.....	72
Listas de argumentos.....	72
Verbos y palabras clave.....	72
Sin efectos secundarios.....	73
Argumentos de salida.....	74
Separación de consultas de comando.....	74

Mejor excepciones que devolver códigos de error.....	75
Extraer bloques Try/Catch .....	76
El procesamiento de errores es una cosa .....	76
El imán de dependencias Error.java .....	76
No repetirse.....	77
Programación estructurada .....	78
Cómo crear este tipo de funciones.....	78
Conclusión .....	78
SetupTeardownInclude .....	79
Bibliografía .....	81
<b>4. Comentarios .....</b>	<b>82</b>
Los comentarios no compensan el código incorrecto.....	84
Explicarse en el código .....	84
Comentarios de calidad .....	85
Comentarios legales .....	85
Comentarios informativos .....	85
Explicar la intención.....	86
Clarificación .....	86
Advertir de las consecuencias .....	87
Comentarios TODO .....	88
Amplificación.....	88
Javadoc en API públicas .....	89
Comentarios incorrectos .....	89
Balbucear .....	89
Comentarios redundantes.....	90
Comentarios confusos.....	92
Comentarios obligatorios .....	92
Comentarios periódicos.....	93
Comentarios sobrantes .....	93
Comentarios sobrantes espeluznantes .....	95
No usar comentarios si se puede usar una función o una variable .....	96
Marcadores de posición.....	96
Comentarios de llave de cierre.....	96
Asignaciones y menciones .....	97
Código comentado .....	97
Comentarios HTML .....	98
Información no local .....	98
Demasiada información .....	99
Conexiones no evidentes.....	99
Encabezados de función .....	100
Javadocs en código no público .....	100
Ejemplo .....	100
Bibliografía .....	103

<b>5. Formato</b> .....	<b>104</b>
La función del formato .....	105
Formato vertical .....	106
La metáfora del periódico .....	107
Apertura vertical entre conceptos.....	107
Densidad vertical.....	108
Distancia vertical .....	109
Declaraciones de variables .....	110
Variables de instancia .....	111
Funciones dependientes .....	112
Afinidad conceptual.....	113
Orden vertical .....	114
Formato horizontal .....	114
Apertura y densidad horizontal.....	115
Alineación horizontal .....	116
Sangrado.....	117
Romper el sangrado .....	119
Ámbitos ficticios.....	119
Reglas de equipo.....	119
Reglas de formato de Uncle Bob .....	120
<b>6. Objetos y estructuras de datos</b> .....	<b>124</b>
Abstracción de datos .....	125
Antisimetría de datos y objetos.....	126
La ley de Demeter .....	129
Choque de trenes.....	129
Híbridos .....	130
Ocultar la estructura .....	131
Objetos de transferencia de datos.....	131
Registro activo .....	132
Conclusión .....	133
Bibliografía.....	133
<b>7. Procesar errores</b> .....	<b>134</b>
Usar excepciones en lugar de códigos devueltos.....	135
Crear primero la instrucción try-catch-finally .....	137
Usar excepciones sin comprobar .....	138
Ofrecer contexto junto a las excepciones .....	139
Definir clases de excepción de acuerdo a las necesidades del invocador.....	139
Definir el flujo normal .....	141
No devolver Null .....	142
No pasar Null .....	143
Conclusión .....	144
Bibliografía.....	145

<b>8. Límites</b> .....	<b>146</b>
Utilizar código de terceros.....	147
Explorar y aprender límites.....	149
Aprender log4j.....	150
Las pruebas de aprendizaje son algo más que gratuitas.....	152
Usar código que todavía no existe.....	152
Límites limpios.....	153
Bibliografía.....	154
<b>9. Pruebas de unidad</b> .....	<b>156</b>
Las tres leyes del DGP.....	158
Realizar pruebas limpias.....	158
Las pruebas propician posibilidades.....	159
Pruebas limpias.....	160
Lenguaje de pruebas específico del dominio.....	163
Un estándar dual.....	163
Una afirmación por prueba.....	165
Un solo concepto por prueba.....	166
F.I.R.S.T.....	167
Conclusión.....	168
Bibliografía.....	168
<b>10. Clases</b> .....	<b>170</b>
Organización de clases.....	171
Encapsulación.....	171
Las clases deben ser de tamaño reducido.....	172
El Principio de responsabilidad única.....	174
Cohesión.....	175
Mantener resultados consistentes en muchas clases de tamaño reducido.....	176
Organizar los cambios.....	182
Aislarnos de los cambios.....	184
Bibliografía.....	186
<b>11. Sistemas</b> .....	<b>188</b>
Cómo construir una ciudad.....	189
Separar la construcción de un sistema de su uso.....	189
Separar Main.....	191
Factorías.....	191
Inyectar dependencias.....	192
Evolucionar.....	193
Aspectos transversales.....	195
Proxies de Java.....	196

Estructuras AOP Java puras .....	198
Aspectos de AspectJ .....	201
Pruebas de unidad de la arquitectura del sistema .....	201
Optimizar la toma de decisiones .....	202
Usar estándares cuando añadan un valor demostrable .....	203
Los sistemas necesitan lenguajes específicos del dominio .....	203
Conclusión .....	204
Bibliografía .....	204
<b>12. Emergencia .....</b>	<b>206</b>
Limpieza a través de diseños emergentes .....	207
Primera regla del diseño sencillo: Ejecutar todas las pruebas .....	208
Reglas 2 a 4 del diseño sencillo: Refactorizar .....	208
Eliminar duplicados .....	209
Expresividad .....	211
Clases y métodos mínimos .....	212
Conclusión .....	212
Bibliografía .....	212
<b>13. Concurrencia .....</b>	<b>214</b>
¿Por qué concurrencia? .....	215
Mitos e imprecisiones .....	216
Desafíos .....	217
Principios de defensa de la concurrencia .....	218
Principio de responsabilidad única (SRP) .....	218
Corolario: Limitar el ámbito de los datos .....	218
Corolario: Usar copias de datos .....	219
Corolario: Los procesos deben ser independientes .....	219
Conocer las bibliotecas .....	219
Colecciones compatibles con procesos .....	220
Conocer los modelos de ejecución .....	220
Productor-Consumidor .....	221
Lectores-Escritores .....	221
La cena de los filósofos .....	222
Dependencias entre métodos sincronizados .....	222
Reducir el tamaño de las secciones sincronizadas .....	223
Crear código de cierre correcto es complicado .....	223
Probar código con procesos .....	224
Considerar los fallos como posibles problemas de los procesos .....	224
Conseguir que primero funcione el código sin procesos .....	225
El código con procesos se debe poder conectar a otros elementos .....	225
El código con procesos debe ser modificable .....	225
Ejecutar con más procesos que procesadores .....	225
Ejecutar en diferentes plataformas .....	226

Diseñar el código para probar y forzar fallos.....	226
Manual .....	226
Automática .....	227
Conclusión .....	228
Bibliografía .....	229
<b>14. Refinamiento sucesivo .....</b>	<b>230</b>
Implementación de Args.....	232
Cómo se ha realizado.....	237
Args: El primer borrador .....	238
Entonces me detuve .....	249
Sobre el incrementalismo .....	249
Argumentos de cadena .....	251
Conclusión .....	288
<b>15. Aspectos internos de JUnit.....</b>	<b>290</b>
La estructura JUnit.....	291
Conclusión .....	304
<b>16. Refactorización de SerialDate .....</b>	<b>306</b>
Primero, conseguir que funcione.....	308
Hacer que sea correcta.....	309
Conclusión .....	322
Bibliografía .....	323
<b>17. Síntomas y heurística .....</b>	<b>324</b>
Comentarios.....	325
C1: Información inapropiada.....	325
C2: Comentario obsoleto .....	326
C3: Comentario redundante .....	326
C4: Comentario mal escrito.....	326
C5: Código comentado .....	326
Entorno .....	327
E1: La generación requiere más de un paso .....	327
E2: Las pruebas requieren más de un paso .....	327
Funciones.....	327
F1: Demasiados argumentos.....	327
F2: Argumentos de salida.....	327
F3: Argumentos de indicador .....	328
F4: Función muerta .....	328
General.....	328
G1: Varios lenguajes en un archivo de código .....	328
G2: Comportamiento evidente no implementado.....	328

G3: Comportamiento incorrecto en los límites .....	329
G4: Medidas de seguridad canceladas .....	329
G5: Duplicación .....	329
G6: Código en un nivel de abstracción incorrecto.....	330
G7: Clases base que dependen de sus variantes.....	331
G8: Exceso de información.....	331
G9: Código muerto.....	332
G10: Separación vertical.....	332
G11: Incoherencia.....	332
G12: Desorden.....	332
G13: Conexiones artificiales.....	333
G14: Envidia de las características.....	333
G15: Argumentos de selector.....	334
G16: Intención desconocida.....	335
G17: Responsabilidad desubicada.....	335
G18: Elementos estáticos incorrectos.....	336
G19: Usar variables explicativas.....	336
G20: Los nombres de función deben indicar lo que hacen.....	337
G21: Comprender el algoritmo.....	337
G22: Convertir dependencias lógicas en físicas.....	338
G23: Polimorfismo antes que If/Else o Switch/Case.....	339
G24: Seguir las convenciones estándar.....	339
G25: Sustituir números mágicos por constantes con nombre.....	339
G26: Precisión.....	340
G27: Estructura sobre convención.....	341
G28: Encapsular condicionales.....	341
G29: Evitar condicionales negativas.....	341
G30: Las funciones solo deben hacer una cosa.....	342
G31: Conexiones temporales ocultas.....	342
G32: Evitar la arbitrariedad.....	343
G33: Encapsular condiciones de límite.....	344
G34: Las funciones solo deben descender un nivel de abstracción.....	344
G35: Mantener los datos configurables en los niveles superiores.....	346
G36: Evitar desplazamientos transitivos.....	346
Java.....	347
J1: Evitar extensas listas de importación mediante el uso de comodines.....	347
J2: No heredar constantes.....	347
J3: Constantes frente a enumeraciones.....	348
Nombres.....	349
N1: Elegir nombres descriptivos.....	349
N2: Elegir nombres en el nivel correcto de abstracción.....	351
N3: Usar nomenclatura estándar siempre que sea posible.....	351
N4: Nombres inequívocos.....	352
N5: Usar nombres extensos para ámbitos extensos.....	352



N6: Evitar codificaciones .....	353
N7: Los nombres deben describir efectos secundarios .....	353
<b>Pruebas (Test) .....</b>	<b>353</b>
T1: Pruebas insuficientes .....	353
T2: Usar una herramienta de cobertura .....	353
T3: No ignorar pruebas triviales .....	354
T4: Una prueba ignorada es una pregunta sobre una ambigüedad .....	354
T5: Probar condiciones de límite .....	354
T6: Probar de forma exhaustiva junto a los errores .....	354
T7: Los patrones de fallo son reveladores .....	354
T8: Los patrones de cobertura de pruebas pueden ser reveladores .....	354
T9: Las pruebas deben ser rápidas .....	355
Conclusión .....	355
Bibliografía .....	355
<b>Apéndices .....</b>	<b>357</b>
<b>Apéndice A Concurrencia II .....</b>	<b>358</b>
Ejemplo cliente/servidor .....	359
El servidor .....	359
Añadir subprocesos .....	361
Observaciones del servidor .....	361
Conclusión .....	363
Posibles rutas de ejecución .....	363
Número de rutas .....	364
Un examen más profundo .....	366
Conclusión .....	368
Conocer su biblioteca .....	369
La estructura Executor .....	369
Soluciones no bloqueantes .....	369
Clases incompatibles con subprocesos .....	371
Las dependencias entre métodos pueden afectar al código concurrente .....	372
Tolerar el fallo .....	373
Bloqueo basado en el cliente .....	373
Boqueo basado en el servidor .....	374
Aumentar la producción .....	376
Cálculo de producción de un solo subproceso .....	377
Cálculo de producción con varios subprocesos .....	377
Bloqueo mutuo .....	378
Exclusión mutua .....	379
Bloqueo y espera .....	379
No expropiación .....	379
Espera circular .....	380
Evitar la exclusión mutua .....	380

Evitar bloqueo y espera .....	380
Evitar la expropiación.....	381
Evitar la espera circular .....	381
Probar código con múltiples subprocesos.....	382
Herramientas para probar código basado en subprocesos.....	385
Conclusión .....	385
Ejemplos de código completos.....	386
Cliente/Servidor sin subprocesos.....	386
Cliente/Servidor con subprocesos.....	389
<b>Apéndice B. org.jfree.date.SerialDate.....</b>	<b>390</b>
<b>Epílogo .....</b>	<b>450</b>
<b>Índice alfabético.....</b>	<b>453</b>