GAÑANADA Nº 008

El parámetro booleano de los siete significados

. . .

CATEGORÍA	ARQUITECTURA CRIMINAL
GRAVEDAD	■■□□□ II — MODERADO
FECHA DEL HALLAZGO	4 DE NOVIEMBRE DE 2026
ETIQUETAS	#diseno-de-apis #legibilidad #booleanos #permisos

§ 1

Descripción del hallazgo

En el servicio de informes de una plataforma de gestión se encontró la siguiente llamada:

```
exportarInforme(informe, true);
```

Nadie sabe qué significa `true` sin abrir la implementación. Y quien abre la implementación tampoco, porque la respuesta es «depende»:

```
export function exportarInforme(informe: Informe, rapido: boolean): Archivo {
  if (rapido && informe.formato === 'pdf') omitirGraficos(informe);
  if (rapido && contexto.esProcesoNocturno) saltarComprobacionDePermisos();
  if (rapido && informe.formato === 'csv') return csvSinCabeceras(informe);
  if (rapido && esViernes()) return enviarPorCorreo(informe);
  // …
}
```

El inventario completo de significados de `rapido === true`, reconstruido a partir del historial:

CONTEXTO	SIGNIFICADO DE `TRUE`	AÑO
Formato PDF	Sin gráficos	2019
Proceso nocturno	Sin comprobar permisos («para que vaya rápido»)	2020
Formato CSV	Sin fila de cabeceras	2020
Viernes	Enviar por correo en lugar de descargar	2021
Usuario administrador	Incluir registros borrados	2022
Informe de más de 10 000 filas	Truncar a 10 000 sin avisar	2023
Llamada desde la API pública	Caché de 24 horas	2025

§ 2

Contexto del descubrimiento

Cada significado se añadió por una razón comprensible y con prisa. Añadir un parámetro nuevo obligaba a cambiar las cuarenta llamadas existentes; reutilizar `rapido`, ninguna. El booleano funcionó como una navaja suiza en la que cada nueva herramienta se encajaba en la hoja que menos se usaba.

El hallazgo se produjo durante una auditoría de seguridad. Un usuario sin permisos sobre un departamento podía descargar sus informes si la exportación coincidía con la ventana del proceso nocturno, porque ese proceso compartía contexto con las peticiones web. La causa raíz era un parámetro de rendimiento que, en un caso particular, desactivaba la autorización.

§ 3

Consecuencias observadas

- **Puntos de llamada ilegibles.** `exportarInforme(informe, true)` no comunica nada; cada lectura exige una visita a la implementación.

- **Acoplamiento entre significados.** Cambiar un comportamiento de `rapido` afecta a los otros seis, porque comparten la misma variable y, a menudo, las mismas ramas.
- **Explosión combinatoria.** Con otros tres booleanos en la misma firma, la función admite dieciséis combinaciones, de las cuales cinco tienen sentido y dos están probadas.
- **Fallo de seguridad.** La autorización quedó condicionada por un parámetro de rendimiento, algo que ningún revisor habría aprobado si se hubiera llamado por su nombre.

§ 4

Recomendación de la academia

Un parámetro booleano suele indicar que la función hace dos cosas. Si hace siete, hace siete.

- **Separa los comportamientos distintos en funciones distintas** cuando lo que cambia es *qué* se hace, no *cómo*.
- **Usa opciones con nombre y tipos específicos** cuando lo que cambia es la configuración:

```
interface OpcionesExportacion {
  graficos: 'incluir' | 'omitir';
  entrega: 'descarga' | 'correo';
  limiteFilas?: number;
}

exportarInforme(informe, { graficos: 'omitir', entrega: 'descarga' });
```

- **Haz explícitas las decisiones en el punto de llamada.** El lector debe entender la llamada sin abrir la función.
- **Nunca condiciones la autorización a un parámetro**, y menos a uno de rendimiento. La comprobación de permisos se aplica siempre, en la capa de entrada, y no admite atajos.

No trabajes así, no seas un gañán.

REFERENCIAS

1 Martin Fowler — «FlagArgument»

Separata del volumen I de la Bitácora del Gañán, publicada por AXXA.
https://axxa.embeddedsource.es/bitacora-del-ganan/008-el-parametro-booleano-de-los-siete-significados