GAÑANADA Nº 009

La variable global que conoce hasta tu vida privada

. . .

CATEGORÍA	SEGURIDAD CREATIVA
GRAVEDAD	■■■■□ IV — EXTREMO
FECHA DEL HALLAZGO	11 DE NOVIEMBRE DE 2026
ETIQUETAS	#estado-global #concurrencia #node-js #privacidad #async

§ 1

Descripción del hallazgo

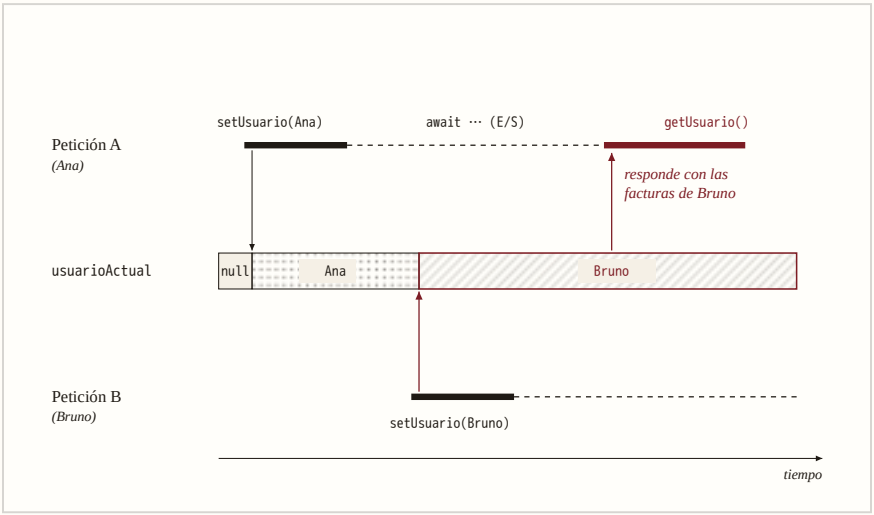
El espécimen habita en el servidor de una aplicación de facturación escrita en Node.js:

```
// contexto.js
let usuarioActual = null;
export const setUsuario = (u) => {
  usuarioActual = u;
};
export const getUsuario = () => usuarioActual;

// app.js
app.use(async (req, res, next) => {
  setUsuario(await sesiones.resolver(req.cookies.sid));
  next();
});

app.get('/api/mis-facturas', async (req, res) => {
  const filtros = await preferencias.cargar(req.query); // ← aquí se cede el
turno
  res.json(await facturas.de(getUsuario().id, filtros));
});
```

Node.js ejecuta el código JavaScript en un solo hilo, y de ahí nace una creencia popular: «no hay problemas de concurrencia». Es falsa. Cada `await` cede el turno al bucle de eventos, y mientras la petición de Ana espera a la base de datos, la de Bruno entra, ejecuta el *middleware* y sobrescribe `usuarioActual`. Cuando la petición de Ana se reanuda, recibe con total amabilidad las facturas de Bruno.



Lám. 9.1 — Cronología de una fuga. Dos peticiones, una variable y un `await` en el peor sitio posible.

§ 2

Contexto del descubrimiento

En el entorno de desarrollo el problema no existe: hay un único usuario, que es quien desarrolla. En preproducción tampoco: las pruebas manuales se hacen de una en una. Los tests automáticos son secuenciales. Todo estaba verde.

En producción, con cientos de peticiones por segundo en horas punta, la ventana entre la asignación y la lectura era suficiente. El hallazgo llegó por la vía más incómoda posible: un cliente escribió, con exquisita

educación, al delegado de protección de datos para preguntar por qué podía ver las facturas de otra empresa.

Lo que siguió fue lo previsto por el RGPD para una violación de la seguridad de los datos personales: evaluación del alcance, registro del incidente y notificación a la autoridad de control sin dilación indebida, en un plazo de 72 horas siempre que sea posible.

§ 3

Consecuencias observadas

- **Fuga de datos entre usuarios**, con obligaciones legales de notificación y una pérdida de confianza difícil de cuantificar.
- **Fallo intermitente y dependiente de la carga**, imposible de reproducir en local y casi imposible de demostrar con los registros habituales.
- **Falsa seguridad en los tests**, que se ejecutaban de uno en uno y no podían ver el problema.
- **Auditoría completa del código**, porque si existía una variable así, nadie podía garantizar que fuera la única.

§ 4

Recomendación de la academia

Los datos de una petición pertenecen a esa petición. Nunca los guardes en estado mutable compartido: variables de módulo, campos estáticos, atributos de un *servlet* o cualquier otro lugar que sobreviva entre peticiones.

La solución más simple es pasar el contexto de forma explícita:

```
app.get('/api/mis-facturas', async (req, res) => {
  const filtros = await preferencias.cargar(req.query);
  res.json(await facturas.de(req.usuario.id, filtros));
});
```

Cuando el contexto debe atravesar muchas capas, Node.js ofrece `AsyncLocalStorage`, que asocia un valor a toda la cadena asíncrona de una petición:

```
import { AsyncLocalStorage } from 'node:async_hooks';

export const contexto = new AsyncLocalStorage();

app.use(async (req, res, next) => {
  const usuario = await sesiones.resolver(req.cookies.sid);
  contexto.run({ usuario }, next);
});

// En cualquier punto de la cadena asíncrona de esa petición:
const { usuario } = contexto.getStore();
```

Y, sobre todo, **prueba el aislamiento con concurrencia real**: un test que lance cientos de peticiones simultáneas de usuarios distintos y compruebe que cada respuesta contiene solo los datos de quien la pidió. Es barato de escribir y habría detectado el espécimen el primer día.

No trabajes así, no seas un gañán.

REFERENCIAS

1 Node.js — «Asynchronous context tracking» (AsyncLocalStorage)

2 Reglamento (UE) 2016/679 (RGPD), artículo 33 — notificación de violaciones de seguridad

Separata del volumen I de la Bitácora del Gañán, publicada por AXXA.
https://axxa.embeddedsource.es/bitacora-del-ganan/009-la-variable-global-que-conoce-hasta-tu-vida-privada