# No todo lo que documenté fue por el sistema

Confesiones de un ingeniero de sistemas en ambientes hostiles

## Prólogo: No era un héroe, solo quería que las cosas funcionaran

Cuando entré al área de sistemas, no sabía que pasaría más tiempo resolviendo política interna que errores de código. Pensé que mi trabajo sería técnico, limpio, lógico. Pronto entendí que escribir software en entornos institucionales es como lanzar botellas al mar… con suerte, una llega.

Este texto no es un manual de buenas prácticas. Es una bitácora de guerra: lo que aprendí, lo que callé, lo que corregí sin pedir permiso y lo que documenté no por el sistema, sino por mí mismo.

## Capítulo 1: Herencia maldita (y otras arquitecturas fantasmas)

Recibí un sistema heredado que nadie entendía, hecho en un lenguaje extinto y sin documentación. El desarrollador anterior había renunciado dejando solo un ZIP y una advertencia: 'No toques el módulo de nómina'.

Pasé semanas descifrando funciones con nombres crípticos. Algunos scripts eran como cartas suicidas de otro tiempo. Aprendí que a veces documentar no es dejar instrucciones, sino evitar maldiciones futuras.

## Capítulo 2: Lo que no dije en la reunión (pero documenté igual)

En cada comité técnico aprendí a callar ciertas cosas. No porque no fueran importantes, sino porque decirlas en voz alta ponía en riesgo el proyecto… o mi permanencia.

Así que creé un archivo paralelo, un 'registro oculto'. Ahí anoté dependencias no declaradas, accesos irregulares, decisiones impuestas por capricho. No por traición, sino por sentido de responsabilidad. Algún día, si algo explotaba, quería dejar constancia de que lo vi venir.

## Capítulo 3: Variables invisibles (y otras amenazas humanas)

El peor bug no está en el código. Está en las relaciones de poder. Una firma que no llega. Un correo ignorado. Un jefe que quiere cambiar el sistema porque lo confundió con una app del banco.

Aprendí a decodificar frases como 'esto debería ser fácil', o 'hazlo como en Google'. Aprendí a blindar entregables con anexos que nadie pedía. No por burocracia, sino para sobrevivir a los cambios de humor institucional.

## Capítulo 4: La épica de los tickets invisibles

Un día normal implica resolver problemas que nunca serán reportados oficialmente. La impresora de la jefa. El acceso a un Excel desde la casa de un funcionario que está 'de descanso activo'.

Esas tareas no cuentan para el sistema. No están en ningún reporte. Pero sostienen la operatividad real de la institución. Y si fallan, no preguntan qué dice el sistema: preguntan qué hiciste tú.

## Capítulo 5: Mi archivo de versiones que no liberé

Tengo una carpeta con funcionalidades completas que nunca se subieron a producción. Algunas fueron canceladas por cambio de prioridades. Otras por miedo. Otras porque funcionaban demasiado bien.

Pero no me arrepiento de haberlas hecho. Me enseñaron cosas que ningún curso de Scrum cubre. Que a veces lo importante no es entregar, sino haber comprendido lo que el sistema no supo pedir.

## Epílogo: El código que aún me pertenece

En este trabajo hay días que terminan en frustración. Pero también hay líneas de código que escribí sabiendo que funcionaban, aunque nadie lo notara. Pequeñas soluciones que evitaron desastres, aunque nunca me lo reconocieran.

No todo lo que documenté fue por el sistema. Lo hice por mí. Para no olvidar que una parte de mí aún cree en hacer las cosas bien, incluso cuando nadie las valida. En un entorno donde todo cambia, eso es lo único que sigue siendo mío.