LITIGIO DE SOFTWARE · PLATAFORMAS DE DESARROLLO

EL SISTEMA FALLABA EN PANTALLA TRAS PANTALLA. Y TODOS LOS ERRORES NACÍAN DEL MOTOR DEL PROVEEDOR.

LA MEJOR FORMA DE ASIGNAR LA CULPA TÉCNICA NO ES DISCUTIR VERSIONES: ES EJECUTAR CADA TAREA Y RASTREAR DÓNDE NACE EL ERROR

> **CUANDO EL MOTOR FALLA, NINGÚN CONSTRUCTOR ARREGLA EL MOTOR.**

Dictamen resuelto clic por clic: la tabla de evidencia que cerró la discusión.

EL PROBLEMA

Una universidad pública contrató la habilitación de una plataforma de gobierno inteligente, y el anexo del contrato definió todo el litigio posterior: la universidad construiría el sistema con sus propios equipos, pero el proveedor entregaría la descripción arquitectónica de cada tarea y, sobre todo, la plataforma de desarrollo sobre la que todo se edificaría.

Cuando el sistema empezó a fallar, la pregunta de los millones fue de quién era la culpa: ¿de los constructores de la universidad o del proveedor de la plataforma?

LA INTERVENCIÓN

El método fue tan simple como contundente: recorrer las tareas generales del sistema y ejecutarlas. Registrar una recepción: error al abrir el documento; la descarga, vacía. Consultar un saldo: el mismo error, la misma descarga vacía. Registrar una entrada de almacén: idéntico resultado. Las validaciones, con reportes sin los campos requeridos y trámites sin información. No era un fallo aislado: era un patrón que recorría el sistema entero.

El análisis llevó a la conclusión que la universidad necesitaba escuchar: esos fallos son propios de la herramienta de desarrollo, no del trabajo de los constructores — el sistema sobre el que construía tenía errores de programación en su núcleo, y era imposible, no difícil, que su personal los solventara. Y había un cerrojo jurídico: el proveedor nunca autorizó modificar la plataforma protegida por derechos de autor. La universidad podía construir sobre ella, pero no repararla: dependía del proveedor hasta para respirar.

Las conclusiones lo asentaron en serie: procesos arquitectónicos incumplidos, plataforma con errores de programación, no funcional ni para construir ni para operar, errores ajenos a los desarrolladores de la universidad, e impacto legal directo en el desarrollo contratado.

Caso anonimizado para proteger la identidad de las partes. Los detalles técnicos se describen a nivel general.

LA PREGUNTA «¿DE QUIÉN ES LA CULPA?» DEJÓ DE SER DISCUSIÓN Y SE CONVIRTIÓ EN TABLA DE EVIDENCIA.

CUANDO EL MISMO FALLO SE REPITE EN TODAS LAS PANTALLAS Y TODOS NACEN DEL NÚCLEO DEL PROVEEDOR

QUÉ SE LOGRÓ

Gracias al dictamen técnico, la universidad documentó en el litigio la asignación exacta de responsabilidad: los errores eran del sistema del proveedor, el desarrollo no podía concluirse en esas condiciones y todo fallo de la plataforma impactaba legalmente en el contrato.

> **La enseñanza para toda institución que compra una plataforma para construir sobre ella: la cláusula que define quién responde por los errores del motor es la más importante del contrato — y se negocia antes de firmar.**

POR QUÉ DURIVA ES LA MEJOR OPCIÓN

≡

EFICIENCIA

Metodología propia probada en miles de dictámenes, con equipos que trabajan en paralelo sobre la evidencia.

TIEMPO

Respuesta ágil en escenarios de urgencia procesal, con la misma calidad técnica y documentación completa.

18

TRAYECTORIA

Más de 18 años de cómputo forense en México y América Latina. Creador del primer sistema operativo forense latinoamericano, presidente del Consejo Americano de Ciberseguridad y Cómputo Forense, autor de seis libros especializados.

Si su sistema falla y nadie sabe de quién es la culpa, ejecútelo tarea por tarea: la evidencia asigna. Somos su aliado digital.

Caso anonimizado para proteger la identidad de las partes. Los detalles técnicos se describen a nivel general.