Engineer

Validación

Comprobar la entrada con reglas antes de confiar en ella

Qué es

La validación es el paso en el que un programa compara los datos entrantes con las reglas que deben cumplir y los rechaza si no las cumplen, antes de guardar, ejecutar o transmitir nada. Recibe distintos nombres según la capa: validación de entrada para los campos de formularios y las peticiones de API, validación de esquema cuando las reglas se expresan como un JSON Schema o una descripción OpenAPI, y validación del marcado cuando se comprueba una página web frente a la gramática de HTML. Una entrada que supera la comprobación se considera válida.

La guía de OWASP divide el trabajo en dos preguntas. La sintaxis comprueba si un valor tiene el tipo y formato esperados: si es una fecha, un entero o un cuerpo que se puede analizar como JSON. La semántica pregunta «si el valor tiene sentido para la operación»: una reserva de hotel necesita dos fechas válidas y, además, una fecha final posterior a la inicial. Convertir la cadena “500” en un entero resuelve la primera pregunta, pero no dice nada sobre si una estancia de 500 noches resulta aceptable.

La validación del marcado es la manifestación pública más antigua de esta idea en la web. W3C Markup Validation Service la describe como comprobar un documento frente a la gramática que afirma usar y compara su validador con lint para C. Su documentación es prudente sobre lo que eso demuestra: una página válida no es necesariamente buena; solo es una que una herramienta ha encontrado conforme con un conjunto concreto de reglas. La misma cautela se aplica a las demás clases. La validación indica que los datos cumplen las reglas escritas, no que se hayan elegido las reglas adecuadas.

A menudo se confunden con ella dos ideas cercanas. El saneamiento modifica la entrada para hacerla aceptable, eliminando o reescribiendo partes, mientras que la validación la acepta o la rechaza tal como llega. La codificación y la parametrización hacen que los datos sean seguros para su destino, sea una página HTML o una consulta SQL, y siguen siendo necesarias después de superar la validación.

Utilice el concepto en una decisión

Aplique «¿Qué es la validación?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Reglas y comprobaciones que detectan datos incorrectos al entrar y aclaran quién responde de cada cifra y de dónde procede.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Un plan de infraestructura para 14 departamentos — 14 departamentos de investigación con un único plan de infraestructura

Un plan de infraestructura para 14 departamentos Póngase en contacto →

Por qué importa

La decisión que sustenta la validación es dónde fija el sistema su límite de confianza y qué comprueba allí. Cada petición de un navegador, mensaje de una cola, archivo de un socio o cambio de configuración de un compañero llega con supuestos asociados. Comprobarlos en la entrada permite que el resto del código confíe en ellos; omitir la comprobación obliga a cada función posterior a defenderse, algo que en la práctica la mayor parte no hará.

La segunda decisión es cómo expresar las reglas. Las comprobaciones escritas a mano, endpoint por endpoint, empiezan siendo baratas y se van separando al crecer el sistema, porque cada una constituye su propia pieza de código. Las reglas declaradas una vez, como tipos o como esquema, pueden hacerse cumplir con una biblioteca y publicarse como documentación desde la misma fuente, lo que reduce el riesgo de que la descripción y el comportamiento se alejen entre sí. El coste es aprender un framework y entender sus opciones predeterminadas: un esquema impone exactamente lo que dice y nada de lo que deja sin decir.

La última elección es la rigidez. Una lista de permitidos describe lo aceptable y rechaza todo lo demás, algo que OWASP recomienda frente a intentar reconocer cada cadena maliciosa. Rechazar campos desconocidos por el esquema detecta pronto las erratas y los clientes desactualizados. Rechazar la petición entera al primer problema, con un error claro, es más seguro que continuar con datos comprobados solo en parte.

Cómo se aplica en la práctica

Una API empezó con validación de peticiones escrita a mano para cada endpoint. En la etapa de implementación documentada, trasladarla a Huma v2 convirtió las comprobaciones en tipos para 649 rutas y 760 operaciones. Cada endpoint tenía estructuras de entrada y salida; Huma generaba la validación de peticiones a partir de ellas, y una descripción OpenAPI salía de los mismos tipos. Dos endpoints WebSocket siguieron usando Fiber directamente, porque el modelo de petición y respuesta no encajaba. Una comprobación de CI comparaba las rutas registradas por el servidor con las anunciadas en la descripción. La paridad de rutas registrada fue del 100 %; las comprobaciones declaradas en los tipos de entrada sustituyeron la validación manual correspondiente.

La configuración requiere el mismo tratamiento: el endurecimiento de SSH en el servidor de la empresa ejecuta ahora tres fases de validación en cada convergencia. Se comprueba la sintaxis del archivo candidato antes de instalarlo, se comprueba la configuración ensamblada después y, por último, se leen los ajustes efectivos del propio demonio y se verifica que cumplan 24 directivas. La tercera fase existe porque un informe de acceso de septiembre de 2026 encontró dos ajustes que la automatización escribía en cada convergencia, pero que el demonio sobrescribía: los valores del propio sistema operativo se leían primero y, para esos dos ajustes, prevalece la primera aparición. Como dice la página, buscar una directiva con grep solo demuestra que se escribió.

La validación del marcado formó parte de la etapa de implementación documentada de agosto de 2026: la validación HTML era una de las comprobaciones de un control de commits con 22 linters sin nivel de advertencias, donde cada diagnóstico era un error y se prohibían las supresiones dentro del código. Por tanto, no se podía hacer commit de una página inválida marcando el error como aceptable.

Dónde surgen los problemas

El servidor confía en lo que ya comprobó el navegador. Un formulario que rechaza un correo incorrecto en el navegador parece validado, pero la petición puede enviarse sin usar el formulario. OWASP señala las comprobaciones exclusivamente del cliente como una trampa habitual: valide en el servidor y trate la comprobación del navegador como una indicación rápida para quien escribe.

Un esquema acepta todo lo que no menciona. En JSON Schema, enumerar una propiedad no la hace obligatoria ni prohíbe otros campos; con el vocabulario predeterminado 2020-12, la palabra clave `format` se recoge como anotación y su comprobación está desactivada de forma predeterminada. Este esquema parece estricto:

```
{
    "type": "object",
    "properties": {
        "email": { "type": "string", "format": "email" }
    }
}
```

Acepta `{}`, `{"email": "nope"}` y `{"emial": "a@example.com"}`. Añada `required` y `"additionalProperties": false`, y active la comprobación del formato en el validador, o compruébelo en el código. Huma, por su parte, rechaza de forma predeterminada los campos desconocidos del cuerpo, documenta `readOnly` y `writeOnly` sin imponerlos e ignora los parámetros de consulta desconocidos salvo que se le indique lo contrario.

Se comprueba una versión de los datos y se usa otra. Un validador que lee la petición sin decodificar mientras el manejador lee una copia decodificada, o una prueba que inspecciona el archivo en disco mientras el demonio aplica otro ajuste, ha comprobado algo que el sistema nunca utiliza. OWASP aconseja: «Valide la representación que se usará realmente». El trabajo de SSH anterior muestra la misma lección aplicada a la configuración: solo la lectura de los ajustes del demonio reveló cuáles estaban en vigor.

Una lista de bloqueados se encuentra con la siguiente variante. Bloquear caracteres problemáticos conocidos rechaza entradas legítimas, como un apóstrofo en un apellido, y aun así deja la consulta expuesta. Defina qué acepta cada campo y rechace el resto.

Se trata una entrada válida como si fuera segura. Una cadena que ha superado todas las reglas aún puede romper una sentencia SQL o una página al pegarla en ellas. La validación limita lo que llega; las consultas parametrizadas y la codificación de salida protegen el lugar donde termina. Se necesitan ambas.

Lecturas adicionales

- OWASP Input Validation Cheat Sheet — sintaxis y semántica, listas de permitidos, límites del análisis y trampas habituales.
- Documentación de W3C Markup Validator — la explicación del propio validador sobre la validación del marcado, con sus páginas de ayuda sobre lo que demuestra y lo que no demuestra la validez.
- JSON Schema Validation, borrador 2020-12 — las palabras clave de validación, incluido el comportamiento predeterminado de `format`.
- Huma: Request Validation — validación generada desde las etiquetas de las estructuras Go, con las opciones estrictas y permisivas para los campos.
- OpenAPI Specification — el formato descriptivo cuyos esquemas aplican muchos validadores de API.

Glosario

Elija según sus restricciones reales

Traiga un conjunto de datos representativo, las preguntas que debe responder y el cuello de botella actual. Podemos acordar cómo medir la calidad, la capacidad de procesamiento y la recuperación.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es la validación?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/validation/