# UD3E7 Validación Progresiva de Formulario con Callbacks

Vamos a implementar un sistema de validación progresiva para un formulario HTML utilizando funciones con callbacks. Cada campo del formulario será validado de manera secuencial y, en caso de error, se detendrá la validación, se resaltará el campo que falló, y se mostrará un mensaje de error informativo al usuario. El objetivo es profundizar en las callbacks y descubrir el "infierno de las callbacks".

El formulario de trabajo incluye los siguientes campos:

- **Nombre**: No puede contener números y debe tener al menos 3 caracteres.
- **Contraseña**: Debe incluir al menos una letra mayúscula, una minúscula, un número, y tener una longitud mínima de 8 caracteres.
- **Email**: Debe contener una única @, texto antes y después de la @, y terminar con un punto seguido de 2 o 3 letras.
- **Fecha de nacimiento**: Validar que el usuario tenga entre 18 y 24 años.

**NOTA:** todas las validaciones se hacen por código sin emplear expresiones regulares.

**Lógica de la aplicación.**

Al pulsar el botón "Guardar", se validarán los campos uno por uno, en el orden indicado.

Cada función validadora debe recibir el valor del campo y un "callback" con dos parámetros: (valor, error).

- En caso de éxito, la función devolverá el valor validado al siguiente paso (siguiente campo a validar).
- En caso de error, la validación se detendrá y se mostrará un mensaje informativo con "alert". En este caso la respuesta de error en el callback será a través de un objeto "ValidacionError". El campo que no supera la validación le aplicamos un estilo para que le ponga borde de color rojo (si el campo correcto este estilo desaparece).

Si todos los campos son válidos se un mensaje con "alert" indicando que el formulario ha sido validado correctamente, y guardaremos los datos del formulario en "**localStorage**".

Al abrir nuestra página verificamos si existen datos en "**localStorage**". Si los hay, precargar los campos del formulario con esos datos.

**Personalización de los errores.**

Vamos a crear un clase "ValidacionError" que extienda "Error" para diferenciar nuestros errores de los errores de otros errores de JavaScript. Esta clase recibe como parámetros el mensaje y el nombre del campo que falló.

**Ampliaciones (en otros ficheros js).**

1. Refactorizar con Promesas

- Reescribir las funciones validadoras para que devuelvan promesas en lugar de usar callbacks.
- Implementar el flujo de validación utilizando ".then()" y ".catch()".

**EVIDENTEMENTE**: las promesas las codificas para que sean asíncronas.

2. Refactorizar con "Async/Await"

- Convertir la lógica del flujo de validación en una función asíncrona utilizando "async" y "await" para mayor legibilidad.