

Guía de Laboratorio De la Semana N°13:

COMPETENCIAS

1. Organización de la estructura de una página web.
2. Agregar interactividad a una página web mediante JavaScript, manipulación del DOM, eventos de ratón y teclado, validación de formularios.

EQUIPOS, MATERIALES, PROGRAMAS Y RECURSOS

- PC Personal.
- Sistema operativo Windows 10.
- Material disponible desde Tecsup Virtual.
- Software requerido.

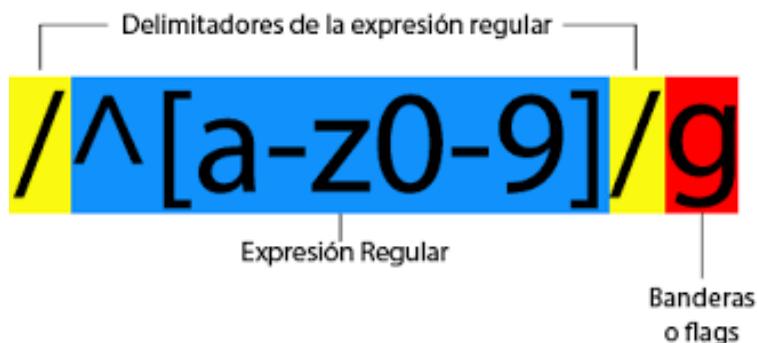
SEGURIDAD

- Colocar las mochilas en el gabinete para evitar caídas en caso de sismo.
- No ingresar con bebidas ni comidas.
- Apagar los equipos y los monitores al culminar la sesión.

PREPARACIÓN

El Alumno debe revisar previamente el material teórico del curso en Tecsup Virtual y revisar su texto.

Expresiones regulares



<https://4geeks.com/es/lesson/expresiones-regulares-javascript>

Una **expresión regular** es una secuencia de caracteres especial que define un patrón de búsqueda en una cadena de texto.

Sirve para:

- Verificar si un texto **cumple ciertas condiciones**
- Buscar ciertos caracteres o combinaciones
- Reemplazar fragmentos de texto

Se usan en **JavaScript, Python, Java**, y casi cualquier lenguaje de programación.

Se colocan entre / /, por ejemplo /\s/ busca espacios en blanco.

Se puede usar con:

- `.test(cadena)` → Devuelve true si encuentra coincidencia.
- `.match(cadena)` → Devuelve las coincidencias.

Símbolo	Significado
.	Cualquier carácter (excepto salto de línea)
^	Inicio de línea
\$	Fin de línea
*	0 o más repeticiones del elemento anterior
+	1 o más repeticiones
?	0 o 1 repetición (opcional)
[]	Conjunto de caracteres [abc] → a, b o c
[^]	Negación de conjunto [^abc] → cualquier cosa menos a, b o c
()	Agrupamiento
\	Escapa caracteres especiales (\ . busca un punto literal)

Ejemplo:

Solo números:

/^[0-9]+\$/

Solo letras minúsculas

/^[a-z]+\$/

Entre 3 y 10 letras, solo letras

/^[a-zA-Z]{3,10}\$/

Entre 2 y 4 dígitos (números)

/^\d{2,4}\$/ →

CASO PRÁCTICO

1. Elabore una página web que cuente con las siguientes características:
 - 1) La web deberá estar conformada por una estructura html, css y javascript en archivos separados y correctamente linkeados (**3 puntos**)
 - 2) La web deberá contar con una barra de navegación con 3 pestañas, la primera llevará a la página de inicio, las otras 2 llevarán a páginas diferentes (recordar que cada página es un archivo html) (**5 puntos**)

- a) Estas páginas deben llevar elementos de texto e imágenes, así como estilos, colores y fuentes diversas.
- 3) En una de las páginas se encontrará un formulario el cual deberá llenar el usuario, este contendrá: **(5 puntos)**
- | | |
|-----------------------|--------------------------|
| a) Nombre | (No números) |
| b) Apellido | (No números) |
| c) DNI | (8 caracteres numéricos) |
| d) Correo electrónico | (debe contener una “@”) |
- 4) Cada dato del formulario deberá ser validado por código javascript usando expresiones regulares para asegurar su correcto llenado **(6 puntos)**

```
assets > js > JS script.js > ...
1   document.getElementById("form").addEventListener("submit", function(event) {
2     event.preventDefault();
3
4     //Código para validación de campos de formulario
5     //...
6     //...
7
8   });
9
10
```

- a) Si los datos ingresados son correctos mostrar un Alert que diga “Datos ingresados con éxito”
- b) Si los datos ingresados no son correctos, deberá mostrar un Alert que diga “Datos Incorrectos, por favor revise los datos ingresados”

Código JavaScript (Captura de pantalla)

```
document.addEventListener('DOMContentLoaded', function() {
    const form = document.getElementById('registroForm');

    const regexNombre = /^[a-zA-ZáéíóúÁÉÍÓÚñÑ\s]+$/;
    const regexApellido = /^[a-zA-ZáéíóúÁÉÍÓÚñÑ\s]+$/;
    const regexDni = /^\d{8}$/;
    const regexEmail = /^[^@\s]+@[^@\s]+\.[^@\s]+$/;

    function validarCampo(input, regex, errorElementId,
mensajeError) {
        const valor = input.value.trim();
        const errorElement =
document.getElementById(errorElementId);

        if (valor === '') {
            errorElement.textContent = 'Este campo es
obligatorio';
            input.classList.add('error');
            return false;
        }

        if (!regex.test(valor)) {
            errorElement.textContent = mensajeError;
            input.classList.add('error');
            return false;
        }

        errorElement.textContent = '';
        input.classList.remove('error');
        return true;
    }

    document.getElementById('nombre').addEventListener('blur',
function() {
    validarCampo(
        this,
        regexNombre,
        'errorNombre',
        'El nombre solo debe contener letras'
    );
});
```

```
document.getElementById('apellido').addEventListener('blur',
function() {
    validarCampo(
        this,
        regexApellido,
        'errorApellido',
        'El apellido solo debe contener letras'
    );
});

document.getElementById('dni').addEventListener('blur',
function() {
    validarCampo(
        this,
        regexDni,
        'errorDni',
        'El DNI debe contener exactamente 8 dígitos'
    );
});

document.getElementById('email').addEventListener('blur',
function() {
    validarCampo(
        this,
        regexEmail,
        'errorEmail',
        'Ingrese un correo válido (debe contener @)'
    );
});

form.addEventListener('submit', function(e) {
    e.preventDefault();

    const nombre = document.getElementById('nombre');
    const apellido = document.getElementById('apellido');
    const dni = document.getElementById('dni');
    const email = document.getElementById('email');

    const nombreValido = validarCampo(
        nombre,
        regexNombre,
        'errorNombre',
        'El nombre solo debe contener letras'
    );

    const apellidoValido = validarCampo(
```

```
    apellido,
    regexApellido,
    'errorApellido',
    'El apellido solo debe contener letras'
);

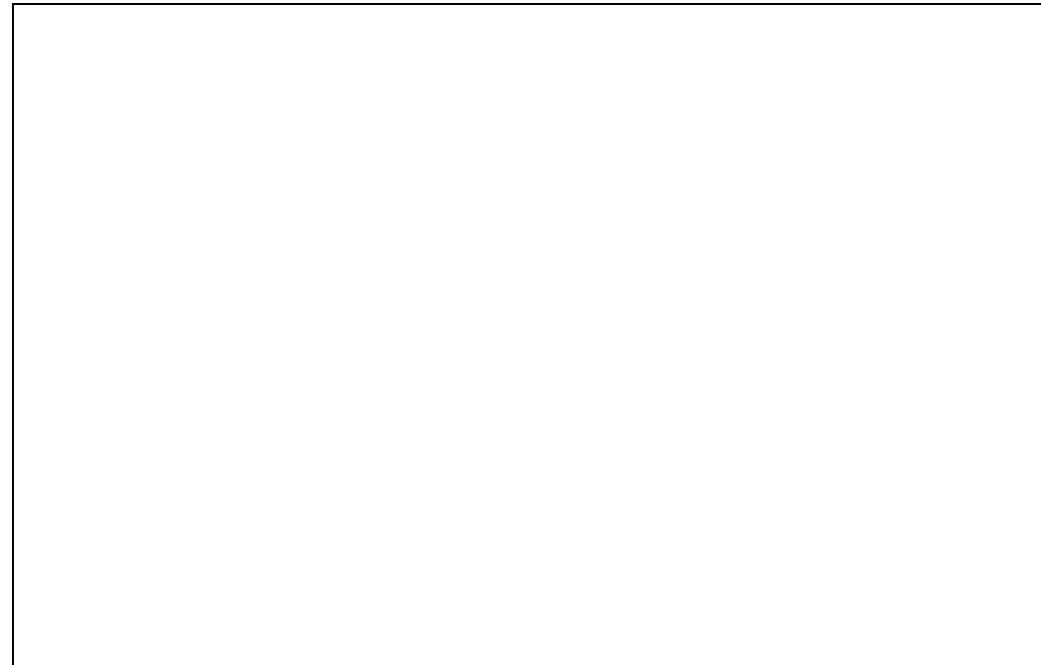
const dniValido = validarCampo(
    dni,
    regexDni,
    'errorDni',
    'El DNI debe contener exactamente 8 dígitos'
);

const emailValido = validarCampo(
    email,
    regexEmail,
    'errorEmail',
    'Ingrese un correo válido (debe contener @)'
);

if (nombreValido && apellidoValido && dniValido &&
emailValido) {
    alert('Datos ingresados con éxito');

    form.reset();
} else {
    alert('Datos Incorrectos, por favor revise los datos
ingresados');
}
});

const inputs = form.querySelectorAll('input');
inputs.forEach(input => {
    input.addEventListener('input', function() {
        this.classList.remove('error');
    });
});
});
```



CONCLUSIONES (Min 5)

Indicar las conclusiones a las que han llegado.

1. Las expresiones regulares son herramientas esenciales para la validación de datos en formularios web, ya que permiten verificar que la información ingresada cumpla con patrones específicos como DNI de 8 dígitos, correos con arroba o nombres sin números, garantizando la integridad de los datos antes de ser procesados.
2. La separación de código en archivos independientes mejora significativamente la organización y mantenibilidad del proyecto, facilitando la identificación de errores, la reutilización de código y permitiendo que múltiples desarrolladores trabajen simultáneamente en diferentes aspectos de la aplicación web.
3. La manipulación del DOM mediante JavaScript permite crear interfaces interactivas y dinámicas, haciendo posible validar datos en tiempo real, mostrar mensajes de error personalizados y mejorar la experiencia del usuario sin necesidad de recargar la página completa.
4. La implementación de una estructura de navegación coherente entre múltiples páginas HTML es fundamental para crear sitios web profesionales, ya que permite a los usuarios moverse fácilmente entre secciones y mantener una experiencia de usuario consistente en todo el sitio.
5. La validación de formularios del lado del cliente mediante JavaScript mejora la experiencia de usuario al proporcionar retroalimentación inmediata sobre errores, reduciendo el tiempo de respuesta y evitando envíos de datos incorrectos al servidor, lo que optimiza tanto el rendimiento como la eficiencia del sistema.