**Examen: Validación y Gestión de Eventos en un Formulario HTML**

**Instrucciones Generales:**

1. Utiliza únicamente el API nativo de JavaScript para resolver este examen.
2. No uses bibliotecas externas como jQuery o frameworks.
3. Debes crear un archivo JavaScript separado llamado formValidation.js y vincularlo al archivo HTML proporcionado.
4. Asegúrate de comentar tu código para explicar las funciones y validaciones que implementas.

**Parte 1: Validación de Campos del Formulario**

Implementa las siguientes validaciones: ­

1. **Nombre (text)**:
   * Este campo es obligatorio y debe contener al menos 3 caracteres.
2. **Correo Electrónico (email)**:
   * Valida que el correo tenga un formato correcto (por ejemplo, usuario@dominio.com).
3. **Contraseña (password)**:
   * Obligatorio, con al menos 8 caracteres, incluyendo al menos una letra mayúscula, una minúscula y un número.
4. **Edad (number)**:
   * Debe ser un número entre 18 y 100.
5. **Fecha de Nacimiento (date)**:
   * Valida que el usuario tenga al menos 18 años.
6. **Teléfono (tel)**:
   * Debe contener solo números y tener una longitud de 10 dígitos.
7. **URL (url)**:
   * Valida que el enlace sea un URL válido y empiece con http:// o https://.
8. **Intereses (checkbox)**:
   * Asegúrate de que al menos un interés esté seleccionado.
9. **País (select)**:
   * Este campo es obligatorio, y el usuario debe elegir una opción que no sea la predeterminada.
10. **Comentarios (textarea)**:
    * Valida que, si el campo se completa, no supere los 200 caracteres.

**Parte 2: Gestión de Eventos**

Implementa las siguientes funcionalidades relacionadas con eventos:

1. **Prevención del envío del formulario:**
   * Si algún campo no pasa la validación, evita que el formulario se envíe y muestra un mensaje de error al usuario.
2. **Eventos en tiempo real:**
   * Valida los siguientes campos mientras el usuario escribe o selecciona valores:
     + Nombre, Correo Electrónico, Contraseña, Edad, Teléfono, y País.
   * Muestra mensajes de validación en tiempo real debajo de los campos.
3. **Manejo de errores globales:**
   * Si el formulario tiene varios errores, muestra un mensaje consolidado al final indicando todos los campos que necesitan correcciones.
4. **Restablecer formulario:**
   * Cuando el usuario haga clic en un botón "Restablecer", limpia todos los campos del formulario y elimina los mensajes de validación.
5. **Mostrar caracteres restantes en tiempo real:**
   * Para el campo "Comentarios", muestra debajo la cantidad de caracteres restantes mientras el usuario escribe. El límite es de 200 caracteres.
6. **Contar campos completados al hacer clic en un botón:**
   * Agrega un botón "Contar Completados". Al hacer clic, muestra un mensaje indicando cuántos campos del formulario han sido completados correctamente.

**Entregables:**

1. Archivo formValidation.js con la solución del examen.
2. Explicación breve (en comentarios del código) de cómo implementaste cada validación y evento.

**Examen: Ejercicios Prácticos de Manejo del DOM con HTML de CV**

**Instrucciones:** Resuelve los siguientes ejercicios utilizando el HTML del CV proporcionado. Usa JavaScript nativo para completar cada tarea.

1. **Selección por ID:** Selecciona la imagen de perfil utilizando su atributo src ("homer\_simpson\_by\_graysonwillis\_d5mrjik-414w-2x.jpg") y cambia su atributo alt a "Foto de perfil actualizada".
2. **Selección por Clase:** Selecciona todos los elementos con la clase contact-info y cambia su color de fondo a un tono azul claro (#e6f7ff). Al hacer click en el H2 de información de contacto.
3. **Selección con querySelector: Usa querySelector para seleccionar el** encabezado principal (h1) y cambia su texto a "Curriculum Vitae Actualizado". Debe ocurrir con el evento mouseover en el elemento <span>HTML</span>. Con el evento mouseout debe volver a su estado original.
4. **Selección con querySelectorAll:** Selecciona todas las etiquetas span dentro de la sección de habilidades (skills) y agrega la clase highlighted-skill a cada una. Debe ocurrir con el evento doble-click sobre el elemento html <span>Figma</span>
5. **Cambio de Contenido con textContent:** Selecciona el primer elemento de la lista en la sección de experiencia laboral y cambia su descripción a "Gestión de proyectos avanzados en desarrollo web.". Debe ocurrir con el evento click sobre el elemento html <span>CSS</span>
6. **Cambio de Contenido con innerHTML:** Selecciona el último elemento de la lista en la sección de educación y reemplaza su contenido con:

<strong>Curso de Especialización en JavaScript</strong> - Academia Profesional (2022) Debe ocurrir con el evento click sobre el elemento html <span>JavaScript</span>

1. **Cambio de Atributos con setAttribute:** Selecciona el enlace de LinkedIn en la sección de contacto y cambia su URL (href) a <https://linkedin.com/in/actualizado>. Debe ocurrir con el evento mouseover sobre el elemento html <span>React</span>
2. **Cambio de Estilos con style:** Selecciona el contenedor principal (.cv-container) y cambia su color de borde a verde (#00ff00).
3. **Manipulación de Clases con classList:** Selecciona el encabezado de la sección de habilidades (skills) y agrega la clase section-highlight. Luego, elimina esta clase después de 3 segundos.
4. **Agregar Elemento con appendChild:** Crea un nuevo elemento li dentro de la sección de experiencia laboral con el contenido:

<strong>Freelancer</strong> - Proyectos Independientes (2023 - Presente)

<p>Desarrollo de soluciones personalizadas para clientes internacionales.</p>

1. **Eliminar Elemento con removeChild:** Selecciona el primer elemento de la lista en la sección de experiencia laboral y elimínalo.
2. **Asociar Evento de Clic:** Agrega un evento de clic al encabezado principal (h1) que cambie su color de texto a rojo al hacer clic.
3. **Prevenir la Acción por Defecto de un Enlace:** Selecciona el enlace de GitHub y prevén que se abra la página al hacer clic, mostrando un mensaje en la consola en su lugar.
4. **Detener la Propagación de Evento:** Agrega un evento de clic a cada div dentro de contact-info que registre un mensaje en la consola. Luego, asegúrate de que, al hacer clic en un div, el evento no se propague al contenedor principal.
5. **Delegación de Eventos:** Agrega un único evento de clic al contenedor contact-info que detecte si se hizo clic en un enlace (a) y muestre un mensaje indicando el texto del enlace clicado.

**Entregables:**

1. Archivo manejoDom.js con la solución del examen.
2. Explicación breve (en comentarios del código) de cómo implementaste cada validación y evento.

# Rúbrica de Evaluación

## **Parte 1: Validación de Campos del Formulario (50 puntos)**

|  |  |  |
| --- | --- | --- |
| Criterio | Descripción | Puntos |
| **Nombre** | Valida que tenga al menos 3 caracteres. | 5 |
| **Correo Electrónico** | Valida que tenga un formato correcto ([usuario@dominio.com](mailto:usuario@dominio.com)). | 5 |
| **Contraseña** | Mínimo 8 caracteres, con al menos una mayúscula, una minúscula y un número. | 5 |
| **Edad** | Debe ser un número entre 18 y 100. | 5 |
| **Fecha de Nacimiento** | Calcula que el usuario tenga al menos 18 años. | 5 |
| **Teléfono** | Debe contener solo números y tener 10 dígitos. | 5 |
| **URL** | Valida que empiece con "http://" o "https://". | 5 |
| **Intereses** | Al menos una casilla seleccionada. | 5 |
| **País** | No debe ser la opción predeterminada. | 5 |
| **Comentarios** | Si se completa, no debe superar 200 caracteres. | 5 |

## **Parte 2: Gestión de Eventos (30 puntos)**

|  |  |  |
| --- | --- | --- |
| Criterio | Descripción | Puntos |
| **Prevención de envío del formulario** | Si hay errores, el formulario no se envía y muestra mensajes. | 5 |
| **Validación en tiempo real** | Se validan los campos mientras el usuario escribe. | 5 |
| **Manejo de errores globales** | Mensaje consolidado de errores en el formulario. | 5 |
| **Restablecer formulario** | Borra campos y elimina mensajes de validación. | 5 |
| **Mostrar caracteres restantes** | Se actualiza dinámicamente en el campo "Comentarios". | 5 |
| **Contar campos completados** | Muestra la cantidad de campos llenados correctamente. | 5 |

## **Parte 3: Ejercicios de Manejo del DOM (50 puntos)**

|  |  |  |
| --- | --- | --- |
| Criterio | Descripción | Puntos |
| **Selección por ID** | Modifica correctamente el atributo alt de la imagen de perfil. | 5 |
| **Selección por Clase** | Cambia correctamente el fondo de contact-info al hacer clic en el h2. | 5 |
| **querySelector con eventos** | Cambia el texto del h1 en mouseover y lo restaura en mouseout. | 5 |
| **querySelectorAll con doble clic** | Agrega clase highlighted-skill a span dentro de skills. | 5 |
| **Cambio con textContent** | Modifica la descripción de la experiencia laboral al hacer clic en CSS. | 5 |
| **Cambio con innerHTML** | Reemplaza correctamente el contenido de educación con innerHTML. | 5 |
| **Cambio de Atributos con setAttribute** | Modifica el enlace de LinkedIn en mouseover. | 5 |
| **Cambio de Estilos con style** | Modifica el borde de .cv-container a verde. | 5 |
| **Manipulación de Clases con classList** | Agrega clase a skills, la elimina tras 3 segundos. | 5 |
| **Agregar y eliminar elementos** | Agrega y elimina elementos en la sección de experiencia laboral. | 5 |

## **Parte 4: Eventos Avanzados en el DOM (20 puntos)**

|  |  |  |
| --- | --- | --- |
| Criterio | Descripción | Puntos |
| **Evento de clic en h1** | Cambia color de texto a rojo. | 5 |
| **Prevenir acción por defecto** | Bloquea el enlace de GitHub y muestra mensaje en consola. | 5 |
| **Detener propagación de eventos** | Evita propagación de eventos en contact-info. | 5 |
| **Delegación de eventos** | Detecta clic en enlaces dentro de contact-info. | 5 |

Total: **150 puntos**