Logotipo

Descripción generada automáticamenteTexto

Descripción generada automáticamente

**Equipo 8:**

* Marco Alejandro González Mireles
* Mirella Stephania Palomera Gómez
* Roberto Carlos Martinez Aviña
* Yosef Sánchez Gutiérrez

**Profesora:** [Rosalia Iniguez Barajas](https://desarrolloweb.udgvirtual.udg.mx/user/view.php?id=174&course=385)

**Fecha de entrega:** 08/Junio/2025

**Registro de Futbolistas – Validación dinámica con DOM y JavaScript**

**Unidad 1**

**Actividad:** 1.2. Obteniendo elementos HTML para su manipulación

**UNIVERSIDAD DE GUADALAJARA**

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS**

**Introducción**

En esta actividad se desarrolló un formulario dinámico utilizando HTML, CSS y JavaScript para reforzar los conocimientos sobre el DOM (Document Object Model), selectores, eventos y manipulación de estilos mediante código. El formulario simula un sistema de registro de futbolistas, permitiendo validar en tiempo real los datos introducidos por el usuario y destacando visualmente los campos correctos o incorrectos.

El propósito principal fue aplicar los métodos `getElementById` y `querySelector` para seleccionar elementos del documento HTML y manipularlos dinámicamente con JavaScript, validando el formulario sin necesidad de enviar datos al servidor.

**Desarrollo de la actividad**

**Estructura del Formulario HTML**

El formulario creado tiene como objetivo registrar datos relevantes de futbolistas, incluyendo nombre, nacionalidad, edad, descripción, posición en el campo y habilidades específicas. Para ello, se utilizaron tres cajas de texto, un área de texto para descripciones más largas, un selector desplegable para la posición y un conjunto de casillas de verificación para seleccionar habilidades, además de un botón para activar la validación.

El diseño del formulario es claro y sencillo, buscando que el usuario pueda ingresar datos rápidamente y recibir retroalimentación inmediata sobre la validez de cada campo.

Al final de este archivo, adjuntamos la imagen 8 con el contenido completo de nuestro archivo HTML, lo adjuntamos al final para facilitar la lectura del proceso de la actividad y las evidencias de la misma.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

*Imagen 1: Captura del formulario renderizado en el navegador*

**Código HTML**

El código HTML del formulario utiliza etiquetas semánticas como `<input>`, `<textarea>`, `<select>` y `<checkbox>`. Se establecieron atributos `id` y `name` para facilitar la selección de elementos en JavaScript. El botón está configurado para no enviar el formulario automáticamente, sino que su función se controla mediante un evento `click` en JavaScript.

Este enfoque permite controlar la interacción del usuario y validar los campos antes de cualquier posible envío.

**Código CSS**

Para el estilo se utilizó CSS moderno, con una paleta de colores inspirada en el fútbol, predominando tonos verdes, blancos y detalles en azul para botones. El formulario está centrado con bordes redondeados y sombra para un aspecto profesional y agradable a la vista.

Se definieron clases `.error` y `.success` para cambiar dinámicamente el borde de los campos dependiendo de si su contenido es válido o no, ayudando visualmente al usuario a identificar los campos que necesitan corrección.

Texto

El contenido generado por IA puede ser incorrecto.

*Imagen 2: Captura de las clases .error y .success en el archivo css*

**Código JavaScript**

El código JavaScript obtiene los elementos del formulario utilizando `getElementById` para campos individuales y `querySelectorAll` para seleccionar todos los checkboxes con el nombre "habilidades". Al hacer clic en el botón, se ejecuta una función que valida cada campo comprobando si están vacíos o cumplen con los requisitos mínimos.

Para cada campo, si el dato es inválido, se aplica la clase `.error`, y si es válido, se aplica `.success`. También se verifica que al menos una habilidad esté seleccionada, mostrando un mensaje de alerta si no es así. La validación es totalmente del lado del cliente, sin recargar la página ni enviar información a un servidor.

Texto

El contenido generado por IA puede ser incorrecto.

*Imagen 3: Captura del código JavaScript completo*

**Validaciones en acción**

Durante la prueba del formulario, se observaron las siguientes respuestas visuales:

* Si un campo requerido queda vacío, su borde se resalta en rojo y la etiqueta también cambia de color, indicando que es necesario completarlo.
* Cuando el campo tiene un valor válido, se resalta con borde verde para confirmar que el dato es aceptable.
* En caso de no seleccionar ninguna habilidad, el sistema despliega un mensaje emergente (alert) indicando que debe elegirse al menos una.

Tabla

El contenido generado por IA puede ser incorrecto.

*Imagen 4: Captura del formulario con campo(s) en error resaltado(s) en rojo*

*Tabla

El contenido generado por IA puede ser incorrecto.*

*Imagen 5: Captura del formulario con campos correctamente llenados resaltados en verde*

*Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.*

*Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.*

*Imagen 6: Captura del alert cuando no se selecciona ninguna habilidad*

**Funcionalidad completa**

Una vez todos los campos están llenados correctamente y se selecciona al menos una habilidad, al presionar el botón de registro aparece un mensaje confirmando que el futbolista fue registrado exitosamente. Esto brinda una retroalimentación clara y positiva al usuario, asegurando que la información fue validada correctamente.

Texto

El contenido generado por IA puede ser incorrecto.

*Imagen 7: Captura del mensaje “¡Futbolista registrado correctamente!”*

**Conclusión**

Esta actividad nos permitió comprender y aplicar de manera práctica la manipulación dinámica del DOM mediante JavaScript, aprendiendo a validar formularios sin necesidad de enviar datos a un servidor y a proporcionar retroalimentación visual al usuario.

El uso combinado de `getElementById` y `querySelector` facilita la obtención eficiente de elementos HTML para su manipulación. Además, diseñar un formulario con temática futbolística hizo el ejercicio más ameno y permitió combinar creatividad con programación.

**Archivos fuente entregados**

* index.html
* styles.css
* script.js

**Bibliografía**

*Fernández, P. E. (2020). Domine JavaScript. 4a Edición. Madrid, España: RA-MA. Pp. 171–189.*

Texto

El contenido generado por IA puede ser incorrecto.

*Imagen 8: Captura del archivo HTML completo*