

20/10/2025

Actividades 3 Aplicaciones Web

Ciclo Formativo de Grado Superior en SMR

Método de entrega

Todas las entregas deberán además de incluir la carpeta con la página web comprimida (ZIP o RAR), también un documento WORD que explique y responda a cada una de las partes descritas en este documento.

Tenéis además tanto en el aula virtual como en el canal de youtube EurekaKamps los videos tutoriales de las clases para que podáis usar en el proceso de desarrollo.

Introducción

En esta tercera actividad de Aplicaciones Web ampliarás la página web creada en la Actividad 2 añadiendo interactividad mediante JavaScript básico. Trabajarás durante 10 sesiones para integrar funciones dinámicas que mejoren la experiencia del usuario, gestionen eventos y modifiquen el contenido de forma visual y accesible.

Objetivos de aprendizaje

- Comprender la estructura y ejecución de scripts en una página web.
- Usar variables, condicionales, bucles y funciones en JavaScript.
- Manipular el DOM (Document Object Model) para cambiar contenido, estilos y atributos.
- Gestionar eventos del usuario (click, input, submit, etc.).

- Validar formularios y generar mensajes dinámicos. - Aplicar principios de interactividad accesible.

Fundamentos de JavaScript y manipulación del DOM

1. Crea un archivo 'script.js' y vincúlalo al final del 'body'.
2. Crea una función de bienvenida que detecte el nombre del usuario (prompt) y lo muestre en el encabezado con innerText.
3. Inserta dinámicamente la fecha actual en el footer.
4. Cambia el color de fondo de una sección al pasar el ratón (mouseover / mouseout).
5. Crea un botón que alterne el modo claro/oscuro modificando clases CSS (classList.toggle).
6. Documenta cada parte con comentarios en el código y en el Word explicativo.

Inserta aquí tus capturas y explicación:

Archivo js vinculado al body.

```
<script src="._/script.js"></script>
```

Función de bienvenida que detecta el nombre de usuario.

```
if(valorRePass==valorRe2Pass){
    alert("Registro completado");
    localStorage.setItem("usuario", valorReUser);
    const display = document.getElementById("usuarioDisplay");
    if (display) {
        display.textContent = `Bienvenido/a, ${valorReUser}`;
        display.style.position = "fixed";
        display.style.top = "10px";
        display.style.right = "10px";
        display.style.zIndex = "9999";
    }
}
else{
    alert("Las contraseñas no son iguales");
}
}
else{
    alert("Indique el nombre de usuario")
}
localStorage.setItem("usuario", valorReUser);
window.addEventListener("DOMContentLoaded", () => {
    const usuario = localStorage.getItem("usuario");
    if (usuario) {
        document.getElementById("usuarioDisplay").textContent = `Bienvenido/a, ${usuario}`;
    }
});
}
```

```
document.addEventListener("DOMContentLoaded", () => {
    const usuario = localStorage.getItem("usuario");
    const display = document.getElementById("usuarioDisplay");
    if (usuario && display) {
        display.textContent = `Bienvenido/a, ${usuario}`;
        display.style.position = "fixed";
        display.style.top = "10px";
        display.style.right = "10px";
        display.style.zIndex = "9999";
    }
});
```

Fecha con la hora que se actualiza dinámicamente.

```
document.addEventListener("DOMContentLoaded", () => {
  const fecha = new Date();

  const opciones = {
    weekday: "long",
    year: "numeric",
    month: "long",
    day: "numeric"
  };

  const fechaFormateada = fecha.toLocaleDateString("es-ES", opciones);

  const contenedor = document.getElementById("fechaActual");
  if (contenedor) {
    contenedor.textContent = `📅 ${fechaFormateada}`;
    contenedor.style.position = "fixed";
    contenedor.style.bottom = "10px";
    contenedor.style.right = "10px";
    contenedor.style.zIndex = "9999";
  }

  // Función para actualizar la hora cada segundo
  function actualizarFechaHora() {
    const ahora = new Date();
    const opcionesFecha = {
      weekday: "long",
      year: "numeric",
      month: "long",
      day: "numeric"
    };
    const fecha = ahora.toLocaleDateString("es-ES", opcionesFecha);
    const hora = ahora.toLocaleTimeString("es-ES", { hour: "2-digit", minute: "2-digit", second: "2-digit" });

    contenedor.textContent = `📅 ${fecha} ⌚ ${hora}`;
  }

  // Actualizar inmediatamente y luego cada segundo
  actualizarFechaHora();
  setInterval(actualizarFechaHora, 1000);
});
```

Se cambia el color del fondo al pasar por encima con el ratón en una sección.

```
document.addEventListener("DOMContentLoaded", () => {
  const seccion = document.getElementById("section");

  seccion.addEventListener("mouseover", () => {
    seccion.style.backgroundColor = "#7c7a7a46";
  });

  seccion.addEventListener("mouseout", () => {
    seccion.style.backgroundColor = ""; // Vuelve al color original
  });
});

document.addEventListener("DOMContentLoaded", () => {
  const seccion = document.getElementById("article");

  seccion.addEventListener("mouseover", () => {
    seccion.style.backgroundColor = "#7c7a7a46";
  });

  seccion.addEventListener("mouseout", () => {
    seccion.style.backgroundColor = ""; // Vuelve al color original
  });
});

document.addEventListener("DOMContentLoaded", () => {
  const seccion = document.getElementById("aside");

  seccion.addEventListener("mouseover", () => {
    seccion.style.backgroundColor = "#7c7a7a46";
  });

  seccion.addEventListener("mouseout", () => {
    seccion.style.backgroundColor = ""; // Vuelve al color original
  });
});
```

Botón de modo oscuro con local storage para guardar la preferencia en todas las páginas.

```
document.addEventListener("DOMContentLoaded", function () {
  const toggle = document.getElementById("darkModeToggle");

  // Activar/desactivar modo oscuro
  toggle.addEventListener("click", function () {
    document.body.classList.toggle("dark-mode");

    // Guardar preferencia en LocalStorage
    if (document.body.classList.contains("dark-mode")) {
      localStorage.setItem("theme", "dark");
    } else {
      localStorage.setItem("theme", "light");
    }
  });

  // Aplicar preferencia guardada
  if (localStorage.getItem("theme") === "dark") {
    document.body.classList.add("dark-mode");
  }
});
```

Interactividad y eventos

1. Implementa un menú desplegable o hamburguesa que se abra y cierre mediante JS.
2. Añade un slider o galería simple con botones “Anterior/Siguiente”.
3. Crea una barra de progreso o contador que se actualice con un evento (scroll o clics).
4. Muestra alertas o mensajes personalizados según la acción del usuario.
5. Añade efectos suaves de aparición/desaparición (style.opacity, setTimeout, classList).

Inserta aquí tus capturas y explicación:

Menú desplegable de hamburguesa con la navegación a las diferentes páginas.

```
document.addEventListener("DOMContentLoaded", () => {
  const boton = document.getElementById("toggleMenu");
  const menu = document.getElementById("menuDesplegable");

  boton.addEventListener("click", () => {
    // Alterna entre mostrar y ocultar el menú
    if (menu.classList.contains("visible")) {
      menu.classList.remove("visible");
      menu.classList.add("oculto");
    } else {
      menu.classList.remove("oculto");
      menu.classList.add("visible");
    }
  });
});
```

Efectos suaves de la página.

```
document.addEventListener("DOMContentLoaded", () => {
  // Espera un momento antes de mostrar la página suavemente
  setTimeout(() => {
    document.body.classList.add("visible");
  }, 100);
});
```

Slider con botones de anterior y siguiente que enseña 3 imágenes de la carpeta "imágenes".

```

window.addEventListener("scroll", () => {
  const scrollTotal = document.documentElement.scrollHeight - window.innerHeight;
  const scrollActual = window.scrollY;
  const porcentaje = (scrollActual / scrollTotal) * 100;

  const barra = document.getElementById("barraScroll");
  if (barra) {
    barra.style.width = `${porcentaje}%`;
  }
});

document.addEventListener("DOMContentLoaded", () => {
  const imagenes = [
    src = "./imagenes/imagen1.webp",
    src = "./imagenes/imagen2.jpg",
    src = "./imagenes/imagen3.jpg"
  ];

  let indiceActual = 0;
  const imagen = document.getElementById("sliderImage");
  const btnAnterior = document.getElementById("prevBtn");
  const btnSiguiente = document.getElementById("nextBtn");

  function mostrarImagen(indice) {
    imagen.src = imagenes[indice];
  }

  btnAnterior.addEventListener("click", () => {
    indiceActual = (indiceActual - 1 + imagenes.length) % imagenes.length;
    mostrarImagen(indiceActual);
  });

  btnSiguiente.addEventListener("click", () => {
    indiceActual = (indiceActual + 1) % imagenes.length;
    mostrarImagen(indiceActual);
  });

  mostrarImagen(indiceActual);
});

```

Barra de progreso de tipo scroll que se ubica arriba en la página enseñando el progreso.

```

window.addEventListener("scroll", () => {
  const scrollTotal = document.documentElement.scrollHeight - window.innerHeight;
  const scrollActual = window.scrollY;
  const porcentaje = (scrollActual / scrollTotal) * 100;

  const barra = document.getElementById("barraScroll");
  if (barra) {
    barra.style.width = `${porcentaje}%`;
  }
});

```

Alertas de registro completado si se cumplen las condiciones, y si no se cumplen se enseñan alertas según lo que falta.


```

    if(valorRePass==valorRe2Pass){
        alert("Registro completado");
        localStorage.setItem("usuario", valorReUser);
        const display = document.getElementById("usuarioDisplay");
        if (display) {
            display.textContent = `Bienvenido/a, ${valorReUser}`;
            display.style.position = "fixed";
            display.style.top = "10px";
            display.style.right = "10px";
            display.style.zIndex = "9999";
        }
    }
    else{
        alert("Las contraseñas no son iguales");
    }
}
else{
    alert("Indique el nombre de usuario")
}
}

```

Formularios dinámicos y validación

1. Mejora el formulario de contacto de la Actividad 2 validando campos vacíos, formato de correo, etc.
2. Muestra mensajes de error y éxito sin recargar la página.
3. Añade iconos o colores de validación mediante JS.
4. Implementa un contador de caracteres o un mensaje de campo obligatorio.
5. Añade un mensaje final animado de confirmación.
6. (Opcional) Guarda los datos temporalmente en localStorage (nombre, correo).

Inserta aquí tus capturas y explicación:

```
document.addEventListener("DOMContentLoaded", () => {
  const form = document.getElementById("contactoForm");
  const mensajeExito = document.getElementById("mensajeExito");

  const nombre = document.getElementById("nombre");
  const email = document.getElementById("email");

  // ♦ Al cargar la página, rellenar con lo guardado en localStorage
  if (localStorage.getItem("nombre")) {
    nombre.value = localStorage.getItem("nombre");
  }
  if (localStorage.getItem("email")) {
    email.value = localStorage.getItem("email");
  }

  form.addEventListener("submit", (e) => {
    let valido = true;

    // Limpiar errores previos
    document.querySelectorAll(".error").forEach(el => el.textContent = "");

    // Validar nombre
    if (nombre.value.trim() === "") {
      document.getElementById("errorNombre").textContent = "El nombre es obligatorio.";
      valido = false;
    }

    // Validar email
    if (!email.value.includes("@")) {
      document.getElementById("errorEmail").textContent = "El correo debe contener '@'.";
      valido = false;
    }
  })
})
```

```

// Validar asunto
const asunto = document.getElementById("asunto");
if (asunto.value.trim() === "") {
  document.getElementById("errorAsunto").textContent = "El asunto es obligatorio.";
  valido = false;
}

// Validar mensaje
const mensaje = document.getElementById("mensaje");
if (mensaje.value.trim().length < 10) {
  document.getElementById("errorMensaje").textContent = "El mensaje debe tener al menos 10 caracteres.";
  valido = false;
}

if (!valido) {
  e.preventDefault();
  return;
}

// Guardar nombre y correo en LocalStorage
localStorage.setItem("nombre", nombre.value);
localStorage.setItem("email", email.value);

// Evitar recarga y mostrar mensaje animado
e.preventDefault();
mensajeExito.innerHTML = `<span class="icono">✔</span> Mensaje enviado correctamente`;
mensajeExito.classList.add("ok");

// Limpiar formulario (excepto nombre y correo, que se mantienen)
document.getElementById("asunto").value = "";
document.getElementById("mensaje").value = "";

```

```

// Limpiar formulario (excepto nombre y correo, que se mantienen)
document.getElementById("asunto").value = "";
document.getElementById("mensaje").value = "";

// Ocultar después de 5s
setTimeout(() => {
  mensajeExito.classList.remove("ok");
  setTimeout(() => {
    mensajeExito.style.display = "none";
  }, 800);
}, 5000);
});

```

Rúbrica de evaluación (100 puntos)

- A. Integración y funcionamiento del script – 20 pt
- B. Manipulación del DOM – 25 pt
- C. Gestión de eventos y formularios – 25 pt
- D. Accesibilidad e interactividad visual – 20 pt
- E. Calidad del código y documentación – 10 pt

Bonus: +5 pt por uso de localStorage, animaciones JS o integración con APIs públicas.

Entregables

1. Carpeta comprimida (ZIP) con todos los archivos actualizados (index.html, styles.css, script.js, assets...).
2. Documento Word con:
 - Capturas antes y después de aplicar JavaScript.
 - Explicaciones breves de cada script.
 - Reflexión final sobre el aprendizaje de JS.
3. Opcional: publicación en GitHub Pages o Netlify.

Evaluación

Con esta práctica se desarrollan y evidencian parcialmente los criterios a), b), c) d), e), f), g), h), i), j) y k) del RA2.

Instrumentos

- Código funcional y estructurado.
- Documento Word explicativo.

NOTA: Este proceso es el mismo que se está realizando en los en clase y publicado en el aula virtual en videos, por lo que se recomienda ver los videos paso a paso.