



Introducción a los sistemas de computo

Examen Final

Metodología resolución problema

Creación de un formulario

El proyecto tiene como funcionalidad almacenar datos de personas, dando como respuesta si su año de nacimiento es bisiesto y si su mes de nacimiento es par/impar.

Pablo René Lémus García

Contenido

Descripción del problema	1
Autor	¡Error! Marcador no definido.
Algoritmo	¡Error! Marcador no definido.
Diccionario de datos.....	¡Error! Marcador no definido.
Código fuente.....	1
Manual de usuario y paso a paso	7

Descripción del problema

Crear un formulario para generar una frase con nuestro nombre y apellido con los años de vida.

Participantes

Pablo René Lémus García

Código fuente HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Examen Final</title>
  <script
src="https://cdn.jsdelivr.net/npm/dexie@3.2.2/dist/dexie.min.js"></script>
</head>
<body>
  <h1>Nombre: Pablo René Lémus García</h1>
  <h2>Usuario de Github: Lemus365</h2>
  <a href="https://github.com/Lemus365" target="_blank">Link hacia mi perfil
de GitHub</a>

  <h2>Formulario:</h2>
  <input type="text" id="nombre" placeholder="Nombre" />
  <input type="text" id="apellido" placeholder="Apellido" />
  <input type="number" id="anioNacimiento" placeholder="Año de nacimiento"
/>
  <input type="number" id="mesNacimiento" placeholder="Mes de nacimiento (1-
12)" min="1" max="12" />
  <button onclick="agregarPersona()">Guardar</button>

  <h3>Personas Registradas</h3>
  <ul id="listaPersonas"></ul>

  <script type="text/javascript">
    const db = new Dexie("PersonasDB");
    db.version(1).stores({
      personas: '++id,nombre,apellido,anioNacimiento,mesNacimiento'
    });

    function calcularEdad(anioNacimiento) {
      const anioActual = new Date().getFullYear();
```



```
    return anioActual - anioNacimiento;
  }

  function esBisiesto(anio) {
    return (anio % 4 === 0 && anio % 100 !== 0) || (anio % 400 === 0);
  }

  function esMesPar(mes) {
    return mes % 2 === 0;
  }

  async function agregarPersona() {
    const nombre = document.getElementById("nombre").value.trim();
    const apellido = document.getElementById("apellido").value.trim();
    const anioNacimiento =
    parseInt(document.getElementById("anioNacimiento").value);
    const mesNacimiento =
    parseInt(document.getElementById("mesNacimiento").value);

    if (!nombre || !apellido || isNaN(anioNacimiento) ||
    isNaN(mesNacimiento) || mesNacimiento < 1 || mesNacimiento > 12) {
      alert("Por favor completa todos los campos correctamente y asegúrate
      que el mes sea entre 1 y 12.");
      return;
    }

    await db.personas.add({ nombre, apellido, anioNacimiento,
    mesNacimiento });

    document.getElementById("nombre").value = '';
    document.getElementById("apellido").value = '';
    document.getElementById("anioNacimiento").value = '';
    document.getElementById("mesNacimiento").value = '';

    listarPersonas();
  }

  async function eliminarPersona(id) {
    await db.personas.delete(id);
    listarPersonas();
  }

  async function listarPersonas() {
```



Introducción a los sistemas de computo

```
const lista = document.getElementById("listaPersonas");
lista.innerHTML = '';

const personas = await db.personas.toArray();

personas.forEach(persona => {
  const edad = calcularEdad(persona.anioNacimiento);
  const bisiesto = esBisiesto(persona.anioNacimiento) ? "Bisiesto" :
"No bisiesto";
  const mesParidad = esMesPar(persona.mesNacimiento) ? "Par" :
"Impar";

  const li = document.createElement("li");
  li.textContent = `${persona.nombre} ${persona.apellido} - Edad:
${edad} años - Año: ${persona.anioNacimiento} (${bisiesto}) - Mes:
${persona.mesNacimiento} (${mesParidad})`;

  const btnEliminar = document.createElement("button");
  btnEliminar.textContent = "Eliminar";
  btnEliminar.style.marginLeft = "10px";
  btnEliminar.onclick = () => eliminarPersona(persona.id);

  li.appendChild(btnEliminar);
  lista.appendChild(li);
});
}

// Cargar lista al inicio
listarPersonas();
</script>
</body>
</html>
```

CÓDIGO JAVASCRIPT ABAJO



```
let db;

const request = indexedDB.open("FormularioDB", 1);

request.onerror = () => {
  console.error("Error al abrir IndexedDB");
};

request.onsuccess = (event) => {
  db = event.target.result;
  console.log("IndexedDB abierta correctamente");
  habilitarBotones();
};

request.onupgradeneeded = (event) => {
  db = event.target.result;
  db.createObjectStore("usuarios", { keyPath: "id", autoIncrement: true });
};

document.getElementById("registroForm").addEventListener("submit", function
(e) {
  e.preventDefault();

  const nombres = document.getElementById("nombres").value.trim();
  const apellidos = document.getElementById("apellidos").value.trim();
  const anio = parseInt(document.getElementById("anioNacimiento").value);
  const mes = parseInt(document.getElementById("mesNacimiento").value);

  const esBisiesto = (anio % 4 === 0 && anio % 100 !== 0) || (anio % 400 ===
0);
  const esMesPar = mes % 2 === 0;

  const resultado = document.getElementById("resultado");

  const transaccion = db.transaction(["usuarios"], "readwrite");
  const store = transaccion.objectStore("usuarios");
```



```
const datos = {
  nombres,
  apellidos,
  anioNacimiento: anio,
  mesNacimiento: mes,
  esBisiesto,
  esMesPar
};

const agregar = store.add(datos);

agregar.onsuccess = () => {
  if (esBisiesto) {
    resultado.textContent = `Datos guardados. Año bisiesto. Mes ${esMesPar
? "par" : "impar"}`;
  } else {
    resultado.textContent = `Datos guardados. Año no bisiesto. Mes
${esMesPar ? "par" : "impar"}`;
  }
};

agregar.onerror = () => {
  resultado.textContent = "Error al guardar en IndexedDB.";
};
});

function habilitarBotones() {
  document.getElementById("verRegistros").addEventListener("click",
mostrarRegistros);
  document.getElementById("borrarRegistros").addEventListener("click",
borrarRegistros);
}

function mostrarRegistros() {
  const lista = document.getElementById("listaRegistros");
  lista.innerHTML = "";

  const transaccion = db.transaction(["usuarios"], "readonly");
  const store = transaccion.objectStore("usuarios");
  const cursor = store.openCursor();

  cursor.onsuccess = (event) => {
```



```
const puntero = event.target.result;
if (puntero) {
  const data = puntero.value;
  const item = document.createElement("li");
  item.textContent = `${data.nombres} ${data.apellidos} - Año:
${data.anioNacimiento} (${data.esBisiesto ? "bisiesto" : "no bisiesto"}) -
Mes: ${data.mesNacimiento} (${data.esMesPar ? "par" : "impar"})`;
  lista.appendChild(item);
  puntero.continue();
} else {
  if (lista.children.length === 0) {
    const vacio = document.createElement("li");
    vacio.textContent = "No hay registros guardados.";
    lista.appendChild(vacio);
  }
}
};

cursor.onerror = () => {
  document.getElementById("resultado").textContent = "Error al leer los
registros.";
};
}

function borrarRegistros() {
  const transaccion = db.transaction(["usuarios"], "readwrite");
  const store = transaccion.objectStore("usuarios");

  const solicitud = store.clear();

  solicitud.onsuccess = () => {
    document.getElementById("resultado").textContent = "Todos los registros
han sido borrados.";
    document.getElementById("listaRegistros").innerHTML = "";
  };

  solicitud.onerror = () => {
    document.getElementById("resultado").textContent = "Error al borrar los
registros.";
  };
}
```




Manual de usuario y paso a paso

Nombre: Pablo René Lémus García

Usuario de Github: Lemus365

[Link hacia mi perfil de GitHub](#)

Formulario:

Nombre Apellido Año de nacimiento Mes de

Personas Registradas

• René Lémus - Edad: 20 años - Año: 2005 (No bisiesto) - Mes: 4 (Par)

1. Coloque su nombre
2. Coloque su Apellido
3. Coloque su Año nacimiento
4. Coloque su mes de nacimiento
5. El resultado se verá en el cuadro del resultado.
6. Botón de eliminar para borrar los datos de usuario