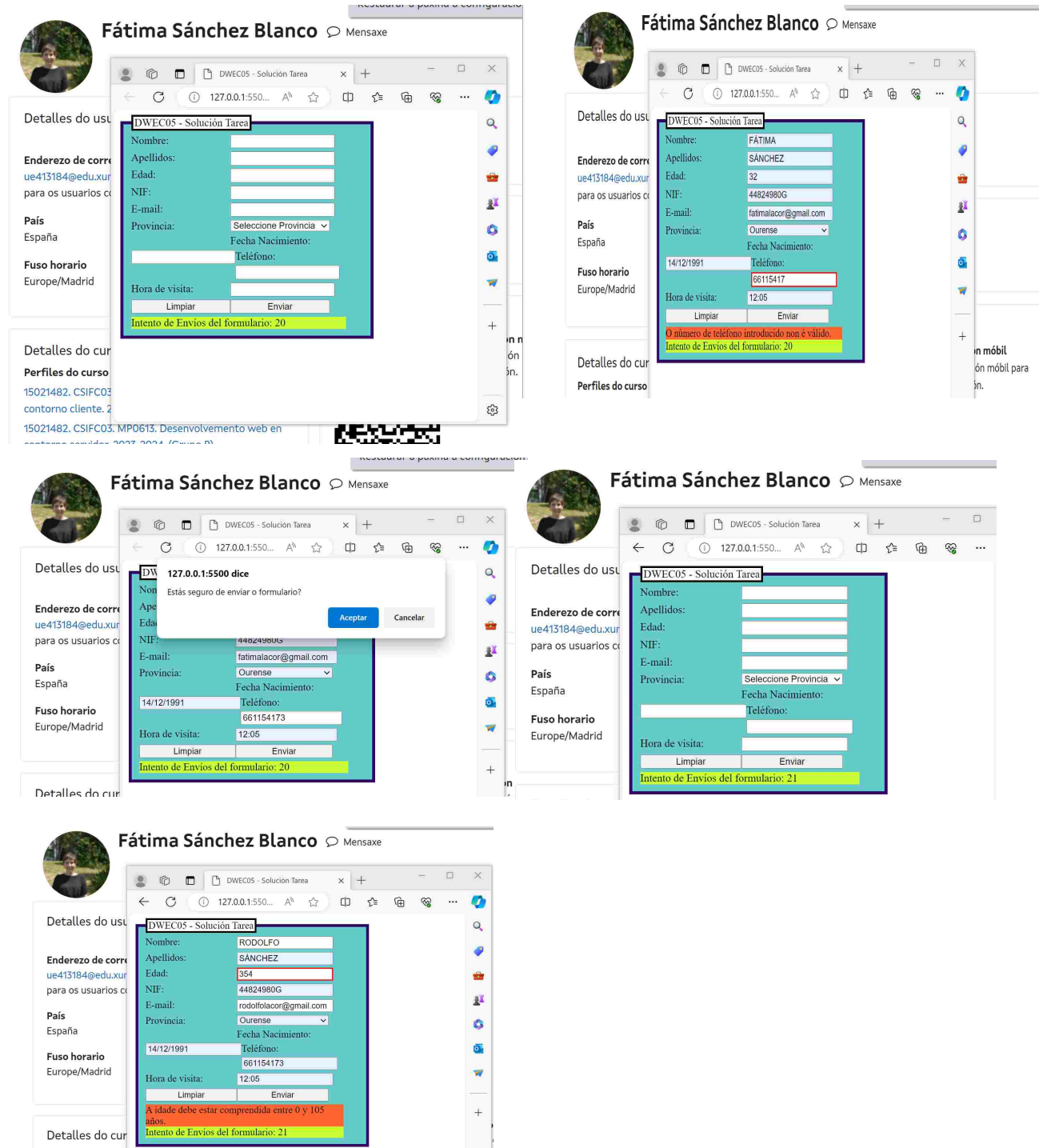


Link ao meu repositório en GitHub:

https://github.com/a22fatimasb/dwcc/tree/main/Tarefa_DWEC05

Demostración:



Código JavaScript:

```
// Función para almacenar e mostrar o número de intentos
function incrementarIntentos() {
    // Obtemos o número de intentos almacenado na cookie, se non hai, comezamos con 0
    let intentos = parseInt(getCookie("intentos")) || 0;
    // Incrementamos o número de intentos en 1
    intentos++;
    // Almacenamos o novo número de intentos na cookie
    setCookie("intentos", intentos, 365);
    // Mostramos o número de intentos actualizado
    mostrarNumeroIntentos();
}

// Función para mostrar o número de intentos no elemento co ID "intentos"
function mostrarNumeroIntentos() {
    let intentos = parseInt(getCookie("intentos")) || 0;
    document.getElementById("intentos").innerHTML =
        "Intento de Envíos del formulario: " + intentos;
}

// Función que crea unha cookie
function setCookie(cnome, cvalor, exdays) {
    // Obtemos a data actual
    let data = new Date();
    // Sumámoslle o número de días dende hoxe para que expire
    data.setTime(data.getTime() + exdays * 24 * 60 * 60 * 1000);
    // Pasamos a data a un formato válido
    let expira = "expires=" + data.toGMTString();
    // Creamos a cookie
    document.cookie = cnome + "=" + cvalor + ";" + expira + ";path=/";
}

// Función para obter unha cookie
function getCookie(nome) {
    // Definimos o prefixo que usaremos para buscar a cookie
    let nomeCookie = nome + "=";
    // Dividimos todas as cookies en partes y decodificamos a cadea de cookies
    let cookies = decodeURIComponent(document.cookie).split(";");
    // Percorremos todas as cookies para buscar a que coincide co nome dado
    for (let i = 0; i < cookies.length; i++) {
        // Eliminamos os espazos en branco ao redor da cookie
        let cookie = cookies[i].trim();
        // Comprobamos se a cookie actual comeza co prefixo do nome buscado
        if (cookie.indexOf(nomeCookie) == 0) {
            // Se coinciden, devolvemos o valor da cookie
            return cookie.substring(nomeCookie.length, cookie.length);
        }
    }
}
```

```

// Se non se atopou a cookie co nome buscado, devolvemos unha cadea baleira
return "";
}

// Función para validar se unha cadea de texto contén só letras
function validarCadeaTexto(cadea) {
    return /^[a-zA-ZáéíóúÁÉÍÓÚ]+$/.test(cadea);
    /* Esta expresión regular indica que a cadea debe comezar (^) con unha serie de
    caracteres do alfabeto, tanto en minúsculas como en maiúsculas, incluíndo as vogais
    con til.
    O (+) establece que debe aparecer polo menos unha vez. E finalmente, o ($) indica
    que a busca se realizará ata o final da cadea. */
}

// Función para validar a idade entre 0 e 105 anos
function validarIdade(idade) {
    return idade >= 0 && idade <= 105;
}

// Función para validar un NIF
function validarDNI(nif) {
    return /^[d{8}[a-zA-Z]$/.test(nif);
    // Expresión que indica que comeza (^) por 8 díxitos (do 0-9) e remata cunha
    letra, tanto en maiúsculas como en minúsculas.
}

// Función para validar un correo electrónico
function validarEmail(email) {
    return /^[^\\s@]+@[^\\s@]+\\.([\\s@]+)$/.test(email);
    /* Expresión que comeza (^) por calquera carácter que non sexa un espazo en branco
    (\\s) nin @, que pode aparecer unha ou máis veces, seguido do carácter @.
    Despois, repetimos a mesma expresión: calquera carácter que non sexa un espazo en
    branco (\\s) nin @, que pode aparecer unha ou máis veces.
    Seguimos cun punto (.), que debe ser escapado (\\.) para ser interpretado
    literalmente.
    Por último, indicamos que a cadea debe rematar coa mesma pauta de calquera
    carácter que non sexa un espazo en branco (\\s) ou @.*/
}

// Función para validar a data no formato dd/mm/aaaa ou dd-mm-aaaa
function validarData(data) {
    return /^(0[1-9]|[12][0-9]|3[01])(-|/)(0[1-9]|1[0-2])(-|/)[d]{4}$/.test(data);
    /* A expresión comeza (^) cun díxito que pode ser 0 seguido doutro díxito do 1 ao
    9, ou ben comeza con 1 ou 2 seguido por un díxito do 0 ao 9.
    Tamén pode comezar con 30 ou 31.
    Despois, debe haber un guión (-) ou unha barra (/) para ser válido.
    A continuación, debe haber un 0 seguido por un díxito do 1 ao 9, ou pode ser un 1
    seguido dun díxito do 0 ao 2.
    Novamente, é necesaria un guión (-) ou unha barra (/) para ser válido.
    E por último, debe rematar con 4 díxitos (do 0 ao 9). */
}

```

```

// Función para validar un número de teléfono de 9 dígitos
function validarTelefono(telefono) {
    return /^d{9}$/.test(telefono);
    // A expresión require que comece (^) e remate ($) cun total de 9 dígitos (\d) (do 0 ao 9).
}

// Función para validar o formato de hora hh:mm
function validarHora(hora) {
    return /^([01]\d|2[0-3]):([0-5]\d)$/.test(hora);
    /* A expresión indica que comece con 0 ou 1 seguido dun dígito (\d) (do 0 ao 9),
    ou pode comezar con 2 seguido dun dígito do 0 ao 3.
    Logo, debe haber dospuntos (:).
    E por último, debe acabar cun dígito do 0 ao 5 seguido de calquera dígito (\d) (do 0 ao 9). */
}

// Función para validar os campos do formulario antes de enviar
function validarFormulario() {
    // Validar nome
    let nome = document.getElementById("nombre").value;
    if (!validarCadeaTexto(nome)) {
        mostrarErros("O nome introducido debe ser unha cadea de texto.");
        document.getElementById("nombre").classList.add("error");
        //document.getElementById("nombre").focus();
        return false;
    }

    // Validar apelidos
    let apelidos = document.getElementById("apellidos").value;
    if (!validarCadeaTexto(apelidos)) {
        mostrarErros("Os apelidos introducidos deben ser unha cadea de texto.");
        document.getElementById("apellidos").classList.add("error");
        //document.getElementById("apellidos").focus();
        return false;
    }

    // Validar idade
    let idade = parseInt(document.getElementById("edad").value);
    if (!validarIdade(idade)) {
        mostrarErros("A idade debe estar comprendida entre 0 y 105 años.");
        document.getElementById("edad").classList.add("error");
        //document.getElementById("edad").focus();
        return false;
    }

    // Validar NIF
    let nif = document.getElementById("nif").value;
    if (!validarDNI(nif)) {
        mostrarErros("O NIF introducido non é válido.");
        document.getElementById("nif").classList.add("error");
        //document.getElementById("nif");
    }
}

```

```

    return false;
}

// Validar email
let email = document.getElementById("email").value;
if (!validarEmail(email)) {
    mostrarErros("O correo electrónico introducido non é válido.");
    document.getElementById("email").classList.add("error");
    //document.getElementById("email").focus();
    return false;
}

// Validar provincia seleccionada
let provincia = document.getElementById("provincia").value;
if (provincia === "0") {
    mostrarErros("Debe seleccionar unha provincia.");
    document.getElementById("provincia").classList.add("error");
    //document.getElementById("provincia").focus();
    return false;
}

// Validar data
let data = document.getElementById("fecha").value;
if (!validarData(data)) {
    mostrarErros("A data debe ter o formato dd/mm/aaaa ou dd-mm-aaaa.");
    document.getElementById("fecha").classList.add("error");
    //document.getElementById("fecha").focus();
    return false;
}

// Validar teléfono
let telefono = document.getElementById("telefono").value;
if (!validarTelefono(telefono)) {
    mostrarErros("O número de teléfono introducido non é válido.");
    document.getElementById("telefono").classList.add("error");
    // document.getElementById("telefono").focus();
    return false;
}

// Validar hora
let hora = document.getElementById("hora").value;
if (!validarHora(hora)) {
    mostrarErros("A hora debe ter o formato hh:mm.");
    document.getElementById("hora").classList.add("error");
    //document.getElementById("hora").focus();
    return false;
}

// Si todas as validacions pasan podese enviar o formulario
return true;
}

```

```

// Función para mostrar mensaxes de erro
function mostrarErros(mensaxe) {
  const errosElemento = document.getElementById("errores");
  if (errosElemento) {
    errosElemento.innerHTML = mensaxe;
  }
}

// Función para convertir a maiúsculas ao perder o foco
function convertirAMaiusculas(elemento) {
  elemento.value = elemento.value.toUpperCase();
}

// Evento que se executa cando o DOM está cargado completamente
document.addEventListener("DOMContentLoaded", function () {
  console.log("DOM cargado");
  // Mostrar o número de intentos ao cargar a páxina
  mostrarNumeroIntentos();

  // Event listener para os campos de nome e apelidos para convertir a maiúsculas
  ao perder o foco
  document.getElementById("nome").addEventListener("blur", function () {
    convertirAMaiusculas(this);
  });

  document.getElementById("apelidos").addEventListener("blur", function () {
    convertirAMaiusculas(this);
  });

  // Event listener para validar o formulario antes de enviar
  document
    .getElementById("formulario")
    .addEventListener("submit", function (event) {
      // Cancelar o envío do formulario por defecto
      event.preventDefault();
      // Validar o formulario antes de enviar
      if (validarFormulario()) {
        let confirmacion = confirm("Estás seguro de enviar o formulario?");
        if (confirmacion) {
          // Si o formulario é válido, incrementar os intentos e enviar o formulario
          incrementarIntentos();
          this.submit(); // Enviar o formulario
        }
      }
    });
});

```