Documentación prueba

Daniel Alejandro Moncaleano Urueña

1.094.954.963

Temporizador

19/01/2023

Clases Creadas

Se crearon las clases:

<div class="imagen" id="cont-princ"></div>

        <div class="columns is-centered">

        <div class="column has-text-centered">

        <div id="contenedorInputs">

        <div class="field">

        <label class="label">Minutos</label>

        <div class="control">

 class="button is-success">

<label class="label">Segundos</label>

El temporizador tiene las siguientes funciones

Inicio = se ingresan los datos dentro de las casillas, el cual <div id="contenedorInputs"> ingresa, al momento de realizar dicha función se ejecuta const iniciarTemporizador = (minutos, segundos) => {

        ocultarElemento($contenedorInputs);

        mostrarElemento($btnPausar);

        ocultarElemento($btnIniciar);

        ocultarElemento($btnDetener);

        if (fechaFuturo) {

            fechaFuturo = new Date(new Date().getTime() + diferenciaTemporal);

            diferenciaTemporal = 0;

        } else {

            const milisegundos = (segundos + (minutos \* 60)) \* 1000;

            fechaFuturo = new Date(new Date().getTime() + milisegundos);

pausa = la app evalua la pausa, en la cual al clickearla guardara el dato en momento, para posteriormente reanudarlo o en su defecto terminarlo en detener const pausarTemporizador = () => {

        ocultarElemento($btnPausar);

        mostrarElemento($btnIniciar);

        mostrarElemento($btnDetener);

        diferenciaTemporal = fechaFuturo.getTime() - new Date().getTime();

        clearInterval(idInterval);

    };

Detener = función final, se encarga de terminar el proceso ejecutado de tal manera que todo quede 00:00

const detenerTemporizador = () => {

        clearInterval(idInterval);

        fechaFuturo = null;

        diferenciaTemporal = 0;

        $tiempoRestante.textContent = "00:00.0";

        init();

    };

Cabe aclarar que la app no funciona si escribimos 0 en minutos y segundos a la vez.

const agregarCeroSiEsNecesario = valor => {

        if (valor < 10) {

            return "0" + valor;

        } else {

            return "" + valor;

        }

    }

    const milisegundosAMinutosYSegundos = (milisegundos) => {

        const minutos = parseInt(milisegundos / 1000 / 60);

        milisegundos -= minutos \* 60 \* 1000;

        segundos = (milisegundos / 1000);

        return `${agregarCeroSiEsNecesario(minutos)}:${agregarCeroSiEsNecesario(segundos.toFixed(1))}`;

    };

    const init = () => {

        $minutos.value = "";

        $segundos.value = "";

        mostrarElemento($contenedorInputs);

        mostrarElemento($btnIniciar);

        ocultarElemento($btnPausar);

        ocultarElemento($btnDetener);

    };

    $btnIniciar.onclick = () => {

        const minutos = parseInt($minutos.value);

        const segundos = parseInt($segundos.value);

        if (isNaN(minutos) || isNaN(segundos) || (segundos <= 0 && minutos <= 0)) {

            return;

        }

        iniciarTemporizador(minutos, segundos);

    };

    init();

    $btnPausar.onclick = pausarTemporizador;

    $btnDetener.onclick = detenerTemporizador;

});