**Código html:**

<!DOCTYPE html>

<html lang="en">

    <li>JUEGO CREATIVO, DIVIERTETE!</li>

<ul>

<li><a href="pokemon-game/index.html">JUEGO</a></li>

</ul>

</li>

<head>

    <BODY background="fondo.jfif">

    </HEAD>

    <BODY >

    <h1><div align="center">ESCUELA TECNOLOGICA DE ESTUDIOS SUPERIORES DE ECATEPEC</div></h1>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>POKEDEX</title>

    <link href="https://fonts.googleapis.com/css2?family=Roboto+Mono&display=swap" rel="stylesheet">

    <script src="script.js" type="text/javascript" defer></script>

    <link href="styles.css" rel="stylesheet" />

</head>

<body>

    <form action="" onsubmit="searchPokemon(event)">

        <input type="text" name="pokemon" autocomplete="off">

    </form>

    <div data-poke-card class="poke-card">

        <div data-poke-name>Pokedex</div>

        <div data-poke-img-container class="img-container">

            <img data-poke-img class="poke-img" src="./dos.jfif"/>

        </div>

        <div data-poke-id> </div>

        <div data-poke-types class="poke-types"></div>

        <div data-poke-stats class="poke-stats"></div>

    </div>

</body>

</html>

**Código java script:**

const pokeCard = document.querySelector('[data-poke-card]');

const pokeName = document.querySelector('[data-poke-name]');

const pokeImg = document.querySelector('[data-poke-img]');

const pokeImgContainer = document.querySelector('[data-poke-img-container]');

const pokeId = document.querySelector('[data-poke-id]');

const pokeTypes = document.querySelector('[data-poke-types]');

const pokeStats = document.querySelector('[data-poke-stats]');

const typeColors = {

    electric: '#FFEA70',

    normal: '#B09398',

    fire: '#FF675C',

    water: '#0596C7',

    ice: '#AFEAFD',

    rock: '#999799',

    flying: '#7AE7C7',

    grass: '#4A9681',

    psychic: '#FFC6D9',

    ghost: '#561D25',

    bug: '#A2FAA3',

    poison: '#795663',

    ground: '#D2B074',

    dragon: '#DA627D',

    steel: '#1D8A99',

    fighting: '#2F2F2F',

    default: '#2A1A1F',

};

const searchPokemon = event => {

    event.preventDefault();

    const { value } = event.target.pokemon;

    fetch(`https://pokeapi.co/api/v2/pokemon/${value.toLowerCase()}`)

        .then(data => data.json())

        .then(response => renderPokemonData(response))

        .catch(err => renderNotFound())

}

const renderPokemonData = data => {

    const sprite =  data.sprites.front\_default;

    const { stats, types } = data;

    pokeName.textContent = data.name;

    pokeImg.setAttribute('src', sprite);

    pokeId.textContent = `Nº ${data.id}`;

    setCardColor(types);

    renderPokemonTypes(types);

    renderPokemonStats(stats);

}

const setCardColor = types => {

    const colorOne = typeColors[types[0].type.name];

    const colorTwo = types[1] ? typeColors[types[1].type.name] : typeColors.default;

    pokeImg.style.background =  `radial-gradient(${colorTwo} 33%, ${colorOne} 33%)`;

    pokeImg.style.backgroundSize = ' 5px 5px';

}

const renderPokemonTypes = types => {

    pokeTypes.innerHTML = '';

    types.forEach(type => {

        const typeTextElement = document.createElement("div");

        typeTextElement.style.color = typeColors[type.type.name];

        typeTextElement.textContent = type.type.name;

        pokeTypes.appendChild(typeTextElement);

    });

}

const renderPokemonStats = stats => {

    pokeStats.innerHTML = '';

    stats.forEach(stat => {

        const statElement = document.createElement("div");

        const statElementName = document.createElement("div");

        const statElementAmount = document.createElement("div");

        statElementName.textContent = stat.stat.name;

        statElementAmount.textContent = stat.base\_stat;

        statElement.appendChild(statElementName);

        statElement.appendChild(statElementAmount);

        pokeStats.appendChild(statElement);

    });

}

const renderNotFound = () => {

    pokeName.textContent = 'No encontrado';

    pokeImg.setAttribute('src', 'poke-shadow.png');

    pokeImg.style.background =  '#fff';

    pokeTypes.innerHTML = '';

    pokeStats.innerHTML = '';

    pokeId.textContent = '';

}

**Código css:**

h1{

    color: #fffb00;

}

form {

    width: 200px;

    margin: 20px auto;

}

input {

    width: 100%;

    padding: 10px;

}

.poke-card {

    position: relative;

    height: fit-content;

    font-family: 'Roboto Mono', monospace;

    max-width: 300px;

    border-radius: 4px;

    color: #000;

    text-align: center;

    padding: 10px;

    margin: 0 auto;

    background-color: #44e9ff;

    border: 1px solid black;

}

.poke-card::before {

    content: '';

    background: radial-gradient(black 33%, transparent 33%);

    background-size: 3px 3px;

    border-radius: 4px;

    height: 100%;

    width: 100%;

    position: absolute;

    left: 10px;

    top: 10px;

    z-index: -1;

}

.img-container {

    position: relative;

    width: 180px;

    margin: 10px auto;

}

.poke-stats div {

    display: flex;

    justify-content: space-between;

    align-content: space-between;

    padding: 5px;

    font-size: 18px;

}

.poke-types div {

    padding: 5px;

    margin: 5px;

    border-radius: 4px;

    border: 1px dashed black;

}

.poke-img {

    width: 200px;

    border-radius: 50%;

}

**Desarrollo Reporte pokedex:**

**Html:**

Primero que nada cree una etiqueta Li para colocar el vínculo del juego creativo que añadí a la página del pokédex, después con un href, di referencia al vínculo de la carpeta donde se encuentra el juego y con una diagonal para que abra el Index directamente del juego, y cómo nombre del vínculo le asigne JUGAR, después metí un background para el fondo de la página del HTML, agregue un título y le di color desde el css haciendo referencia, en el archivo html hacemos referencia a los archivos que ocuparemos, en este caso son los archivos css, js, agregamos un form action onsubmite para la búsqueda del Pokémon que en este caso sería nuestra función, vamos a ocupar los elementos data donde va a ir el contenido de la card (pokedex), en cuanto al siguiente data poke-card class es donde colocaremos el nombre del Pokémon, después colocaremos la imagen del Pokémon haciendo referencia al link donde se encuentra, también vamos a ocupar un data para el id del Pokémon

Captura de pantalla de computadora

Descripción generada automáticamente

**Java script:**

En cuanto al archivo js vamos a crearlo y vamos a seleccionar los data que habíamos mencionado anteriormente en el archivo HTML, después se creó un diccionario para posicionar los colores esperados del Pokémon mapeándolo con los códigos de color y el tipo de Pokémon.

Mandamos llamar la búsqueda del Pokémon del archivo HTML, después colocamos un fech para traer el api de pokeapi con el link de la misma, después creamos un setCardColor para el fondo de la imagen del Pokémon

Captura de pantalla de computadora

Descripción generada automáticamente

**Css:**

Cómo ya lo había mencionado anteriormente jalamos la etiqueta h1 para cambiar el color del título que en este caso fue, escuela de estudios superiores de Ecatepec en color amarillo, después con un from cambiamos el tamaño y el margen de la letra y colocamos el tipo de letra en automático. proseguimos con el archivo css, colocamos el tamaño del form y el input para que no quede pequeño el centrado del buscador, en cuanto al poke-card lo mandamos llamar para modificar la posición la fuente, el tamaño de la fuente, los bordes y el margen de la misma, para que tome forma de la tarjeta del Pokémon, también colocamos el fondo transparente para poder observar la información del Pokémon, también colocamos diseño y tamaño a img-container para posicionar la imagen, el tamaño y el margen

Captura de pantalla de computadora

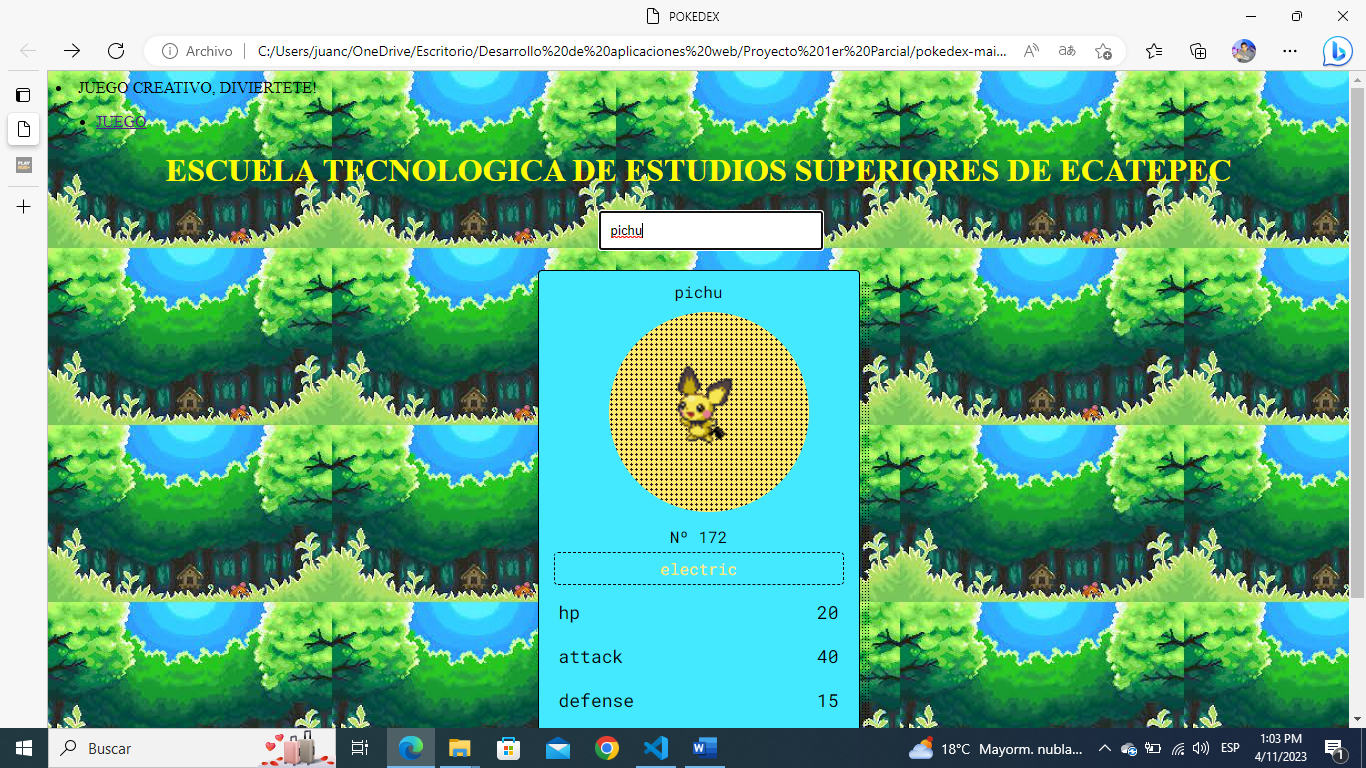
Descripción generada automáticamente

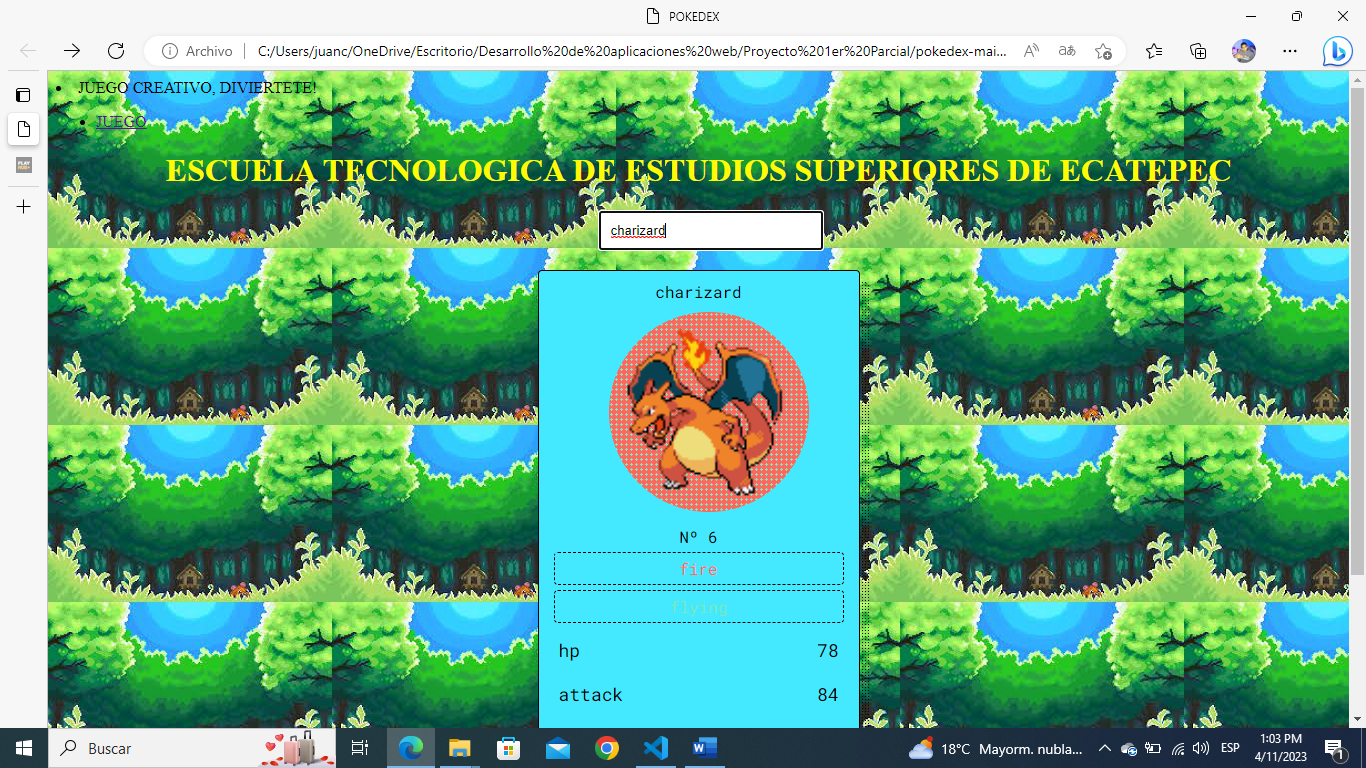
**Capturas de resultado final:**

**Inicio:**



**Resultado de búsqueda:**







**JUEGO DE POKEMON-GAME**

**Codigo html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>POKEMON JUEGO BASICO</title>

**Codigo css:**

    <style>

\* {

    padding: 0;

    margin: 0;

}

body{

    height: 100vh;

    background: #584040;

    display: flex;

    align-items: center;

}

.contenedor {

    width: 920px;

    height: 280px;

    margin: 0 auto;

    position: relative;

    background: linear-gradient(#b7d6c7, transparent) #ffe2d1;/\*linear-gradient(#90ebff, white);\*/

    transition: background-color 1s linear;

    overflow: hidden;

}

.mediodia {

    background-color: #ffdcf3;

}

.tarde {

    background-color: #ffadad;

}

.noche {

    background-color: #aca8c7;

}

.pika {

    width: 84px;

    height: 84px;

    position: absolute;

    bottom: 22px;

    left: 42px;

    z-index: 2;

    background: url(img/pikachu.png) repeat-x 0px 0px;

    background-size: 336px 84px;

    background-position-x: 0px;

}

.pika-corriendo {

    animation: animarDino 0.25s steps(2) infinite;

}

.pika-estrellado {

    background-position-x: -252px;

}

.suelo {

    width: 200%;

    height: 42px;

    position: absolute;

    bottom: 0;

    left: 0;

    background: url(img/suelo.png) repeat-x 0px 0px;

    background-size: 50% 42px;

}

.fuego{

    width: 46px;

    height: 96px;

    position: absolute;

    bottom: 16px;

    left: 600px;

    z-index: 1;

    background: url(img/fuego.png) no-repeat;

}

.arbusto{

    width: 98px;

    height: 66px;

    background: url(img/arbusto.png) no-repeat;

}

.nube{

    width: 92px;

    height: 26px;

    position: absolute;

    z-index: 0;

    background: url(img/nube.png) no-repeat;

    background-size: 92px 26px;

}

.score{

    width: 100px;

    height: 30px;

    position: absolute;

    top: 5px;

    right: 15px;

    z-index: 10;

    color: #e0ff34;

    font-family: Verdana;

    font-size: 30px;

    font-weight: bold;

    text-align: right;

}

.game-over{

    display: none;

    position: absolute;

    width: 100%;

    text-align: center;

    color: #7e928b;

    font-size: 30px;

    font-family: Verdana;

    font-weight: 700;

}

@keyframes animarpika{

    from{

        background-position-x: -84px;

    }

    to{

        background-position-x: -252px;

    }

}

    </style>

</head>

<body>

    <div class="contenedor">

        <div class="suelo"></div>

        <div class="pika pika-corriendo"></div>

        <div class="score">0</div>

    </div>

    <div class="game-over">GAME OVER</div>

<script>

//\*\*\*\*\*\* GAME LOOP \*\*\*\*\*\*\*\*//

var time = new Date();

var deltaTime = 0;

if(document.readyState === "complete" || document.readyState === "interactive"){

    setTimeout(Init, 1);

}else{

    document.addEventListener("DOMContentLoaded", Init);

}

function Init() {

    time = new Date();

    Start();

    Loop();

}

function Loop() {

    deltaTime = (new Date() - time) / 1000;

    time = new Date();

    Update();

    requestAnimationFrame(Loop);

}

**Codigo JS**

//\*\*\*\*\*\* GAME LOGIC \*\*\*\*\*\*\*\*//

var sueloY = 22;

var velY = 0;

var impulso = 900;

var gravedad = 2500;

var pikaPosX = 42;

var pikaPosY = sueloY;

var sueloX = 0;

var velEscenario = 1280/3;

var gameVel = 1;

var score = 0;

var parado = false;

var saltando = false;

var tiempoHastaObstaculo = 2;

var tiempoObstaculoMin = 0.7;

var tiempoObstaculoMax = 1.8;

var obstaculoPosY = 16;

var obstaculos = [];

var tiempoHastaNube = 0.5;

var tiempoNubeMin = 0.7;

var tiempoNubeMax = 2.7;

var maxNubeY = 270;

var minNubeY = 100;

var nubes = [];

var velNube = 0.5;

var contenedor;

var pika;

var textoScore;

var suelo;

var gameOver;

function Start() {

    gameOver = document.querySelector(".game-over");

    suelo = document.querySelector(".suelo");

    contenedor = document.querySelector(".contenedor");

    textoScore = document.querySelector(".score");

    pika = document.querySelector(".pika");

    document.addEventListener("keydown", HandleKeyDown);

}

function Update() {

    if(parado) return;

    Moverpika();

    MoverSuelo();

    DecidirCrearObstaculos();

    DecidirCrearNubes();

    MoverObstaculos();

    MoverNubes();

    DetectarColision();

    velY -= gravedad \* deltaTime;

}

function HandleKeyDown(ev){

    if(ev.keyCode == 32){

        Saltar();

    }

}

function Saltar(){

    if(pikaPosY === sueloY){

        saltando = true;

        velY = impulso;

        pika.classList.remove("pika-corriendo");

    }

}

function Moverpika() {

    pikaPosY += velY \* deltaTime;

    if(pikaPosY < sueloY){

        TocarSuelo();

    }

    pika.style.bottom = pikaPosY+"px";

}

function TocarSuelo() {

    pikaPosY = sueloY;

    velY = 0;

    if(saltando){

        pika.classList.add("pika-corriendo");

    }

    saltando = false;

}

function MoverSuelo() {

    sueloX += CalcularDesplazamiento();

    suelo.style.left = -(sueloX % contenedor.clientWidth) + "px";

}

function CalcularDesplazamiento() {

    return velEscenario \* deltaTime \* gameVel;

}

function Estrellarse() {

    pika.classList.remove("pika-corriendo");

    pika.classList.add("pika-estrellado");

    parado = true;

}

function DecidirCrearObstaculos() {

    tiempoHastaObstaculo -= deltaTime;

    if(tiempoHastaObstaculo <= 0) {

        CrearObstaculo();

    }

}

function DecidirCrearNubes() {

    tiempoHastaNube -= deltaTime;

    if(tiempoHastaNube <= 0) {

        CrearNube();

    }

}

function CrearObstaculo() {

    var obstaculo = document.createElement("div");

    contenedor.appendChild(obstaculo);

    obstaculo.classList.add("fuego");

    if(Math.random() > 0.5) obstaculo.classList.add("arbusto");

    obstaculo.posX = contenedor.clientWidth;

    obstaculo.style.left = contenedor.clientWidth+"px";

    obstaculos.push(obstaculo);

    tiempoHastaObstaculo = tiempoObstaculoMin + Math.random() \* (tiempoObstaculoMax-tiempoObstaculoMin) / gameVel;

}

function CrearNube() {

    var nube = document.createElement("div");

    contenedor.appendChild(nube);

    nube.classList.add("nube");

    nube.posX = contenedor.clientWidth;

    nube.style.left = contenedor.clientWidth+"px";

    nube.style.bottom = minNubeY + Math.random() \* (maxNubeY-minNubeY)+"px";

    nubes.push(nube);

    tiempoHastaNube = tiempoNubeMin + Math.random() \* (tiempoNubeMax-tiempoNubeMin) / gameVel;

}

function MoverObstaculos() {

    for (var i = obstaculos.length - 1; i >= 0; i--) {

        if(obstaculos[i].posX < -obstaculos[i].clientWidth) {

            obstaculos[i].parentNode.removeChild(obstaculos[i]);

            obstaculos.splice(i, 1);

            GanarPuntos();

        }else{

            obstaculos[i].posX -= CalcularDesplazamiento();

            obstaculos[i].style.left = obstaculos[i].posX+"px";

        }

    }

}

function MoverNubes() {

    for (var i = nubes.length - 1; i >= 0; i--) {

        if(nubes[i].posX < -nubes[i].clientWidth) {

            nubes[i].parentNode.removeChild(nubes[i]);

            nubes.splice(i, 1);

        }else{

            nubes[i].posX -= CalcularDesplazamiento() \* velNube;

            nubes[i].style.left = nubes[i].posX+"px";

        }

    }

}

function GanarPuntos() {

    score++;

    textoScore.innerText = score;

    if(score == 5){

        gameVel = 1.5;

        contenedor.classList.add("mediodia");

    }else if(score == 10) {

        gameVel = 2;

        contenedor.classList.add("tarde");

    } else if(score == 20) {

        gameVel = 3;

        contenedor.classList.add("noche");

    }

    suelo.style.animationDuration = (3/gameVel)+"s";

}

function GameOver() {

    Estrellarse();

    gameOver.style.display = "block";

}

function DetectarColision() {

    for (var i = 0; i < obstaculos.length; i++) {

        if(obstaculos[i].posX > pikaPosX + pika.clientWidth) {

            //EVADE

            break; //al estar en orden, no puede chocar con más

        }else{

            if(IsCollision(pika, obstaculos[i], 10, 30, 15, 20)) {

                GameOver();

            }

        }

    }

}

function IsCollision(a, b, paddingTop, paddingRight, paddingBottom, paddingLeft) {

    var aRect = a.getBoundingClientRect();

    var bRect = b.getBoundingClientRect();

    return !(

        ((aRect.top + aRect.height - paddingBottom) < (bRect.top)) ||

        (aRect.top + paddingTop > (bRect.top + bRect.height)) ||

        ((aRect.left + aRect.width - paddingRight) < bRect.left) ||

        (aRect.left + paddingLeft > (bRect.left + bRect.width))

    );

}

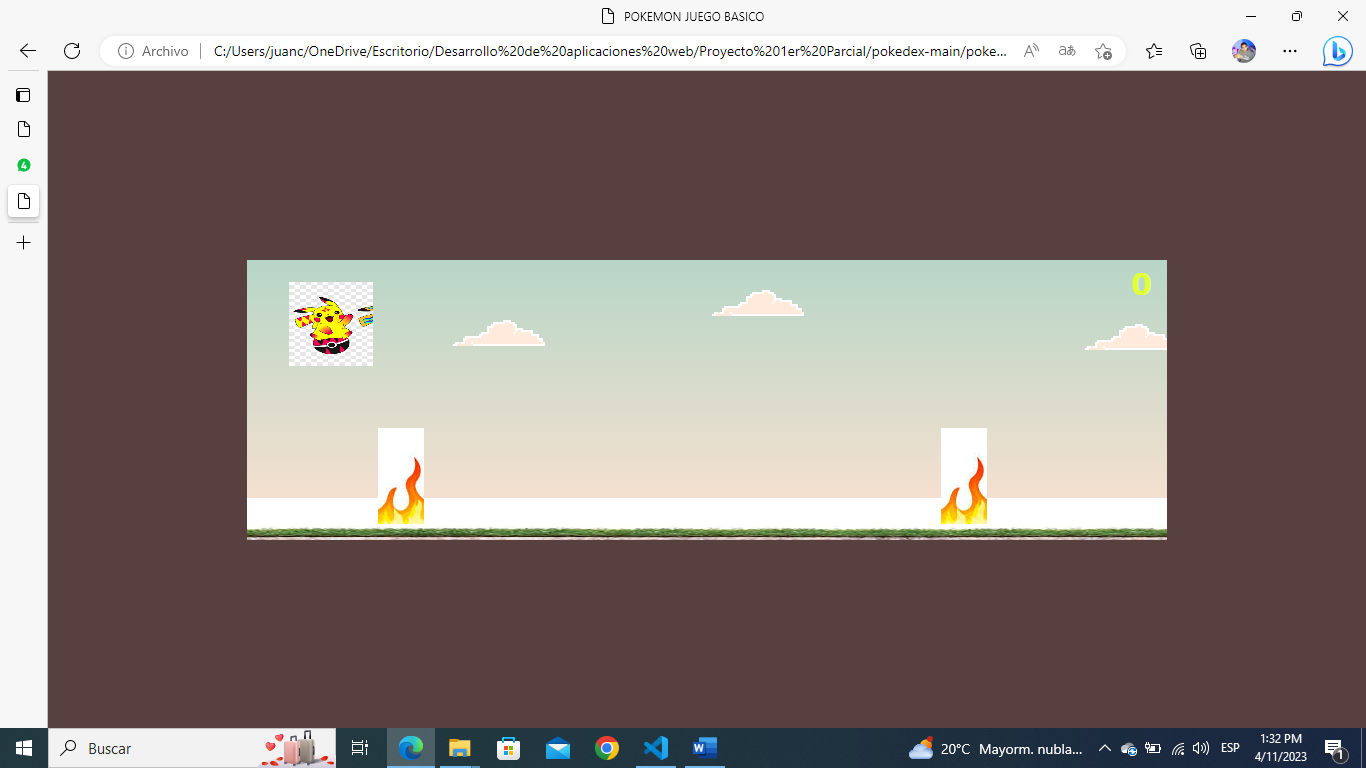
</script>

</body>

</html>

**RESULTADO**

**JUEGO CORRIENDO:**



**FIN DEL JUEGO**

