

Tarea 09 – **Resuelto**

Estimado/a participante,

En esta tarea, deberás resolver una serie de problemas utilizando el lenguaje de programación JavaScript. Para completarla, sigue los siguientes pasos:

1. Crea un repositorio en GitHub donde subirás tu trabajo. Asegúrate de que el repositorio sea público para que podamos revisarlo.
2. Dentro del repositorio, crea un archivo para cada problema que se presenta a continuación. Cada archivo debe tener extensión `.js` y debe contener la solución al problema correspondiente.
3. Además de los archivos `.js`, crea un archivo PDF que contenga todas las soluciones a los problemas. Este archivo debe estar bien estructurado y presentado de manera clara.
4. Una vez que hayas completado todos los pasos, comparte el enlace al repositorio de GitHub en la casilla correspondiente del GES para que podamos revisar tu trabajo.
5. ¡Buena suerte!



Esta foto de Autor
desconocido está bajo
licencia [CC BY-NC](#)

Problema I (15 puntos)

Escribe un programa que lea continuamente el nombre y la distancia de dos enemigos y determine cuál de ellos está más cerca. Imprime el nombre del enemigo más cercano.

Entrada:

enemy1: "EnemyA"
10
enemy2: "EnemyB"
dist2: 20

Salida:

EnemyA dist1:

```
// Función principal
function encontrarEnemigoMasCercano() {
  // Continuarmente leer entradas
  while (true) {
    // Leer los nombres y las distancias de los enemigos
    let enemy1 = prompt("Ingrese el nombre del primer enemigo:");
    let dist1 = parseFloat(prompt("Ingrese la distancia del primer enemigo:"));
    let enemy2 = prompt("Ingrese el nombre del segundo enemigo:");
    let dist2 = parseFloat(prompt("Ingrese la distancia del segundo enemigo:"));

    // Determinar cuál enemigo está más cerca
    if (dist1 < dist2) {
      console.log(enemy1);
    } else {
      console.log(enemy2);
    }
  }
}

// Ejecutar la función principal
encontrarEnemigoMasCercano();
```

Problema II (15 puntos)

En un juego de disparos, tu objetivo es atacar siempre al enemigo más cercano. Escribe un programa que lea continuamente el nombre y la distancia de dos enemigos y seleccione el objetivo más cercano para atacar. Si ambos enemigos están a la misma distancia, selecciona el que aparece primero.

Entrada:

enemy1: "EnemyX"

dist1: 15

enemy2: "EnemyY"

dist2: 15

Salida:

EnemyX

```
// Función principal
function atacarEnemigoMasCercano() {
  // Continuarmente leer entradas
  while (true) {
    // Leer los nombres y las distancias de los enemigos
    let enemy1 = prompt("Ingrese el nombre del primer enemigo:");
    let dist1 = parseFloat(prompt("Ingrese la distancia del primer enemigo:"));
    let enemy2 = prompt("Ingrese el nombre del segundo enemigo:");
    let dist2 = parseFloat(prompt("Ingrese la distancia del segundo enemigo:"));

    // Determinar cuál enemigo atacar
    if (dist1 < dist2) {
      console.log(enemy1);
    } else if (dist2 < dist1) {
      console.log(enemy2);
    } else {
      console.log(enemy1); // Si las distancias son iguales, atacar al primero
    }
  }
}

// Ejecutar la función principal
atacarEnemigoMasCercano();
```

Problema III (20 Puntos)

En una simulación de radar, los datos de enemigos y sus distancias cambian dinámicamente. Escribe un programa que lea continuamente el nombre y la distancia de dos enemigos y determine cuál de ellos está más cerca. Asegúrate de que el programa maneje correctamente entradas donde las distancias pueden cambiar rápidamente. Utiliza prompt para ingresar los datos.

Entrada:

enemy1: "Enemy1"

enemy2: "Enemy2"

dist2: 30

Salida:

Enemy2 dist1: 50

enemy1: "Enemy1"

enemy2: "Enemy2"

dist2: 45

Enemy1 dist1: 25

```
// Función principal
function radarSimulacion() {
  // Continuarmente leer entradas
  while (true) {
    // Leer los nombres y las distancias de los enemigos
    let enemy1 = prompt("Ingrese el nombre del primer enemigo:");
    let dist1 = parseFloat(prompt("Ingrese la distancia del primer enemigo:"));
    let enemy2 = prompt("Ingrese el nombre del segundo enemigo:");
    let dist2 = parseFloat(prompt("Ingrese la distancia del segundo enemigo:"));

    // Determinar cuál enemigo está más cerca
    if (dist1 < dist2) {
      console.log(enemy1);
    } else if (dist2 < dist1) {
      console.log(enemy2);
    } else {
      console.log(enemy1); // Si las distancias son iguales, seleccionar el primero
    }

    // Esperar un corto período de tiempo antes de la siguiente lectura
    setTimeout(() => {}, 500); // 500 ms de espera
  }
}

// Ejecutar la función principal
radarSimulacion();
```

Problema IV (20 Puntos)

En una situación de combate, además de la distancia, se te proporciona la prioridad de ataque de cada enemigo. Escribe un programa que lea continuamente el nombre, la distancia y la prioridad de dos enemigos y seleccione el enemigo con mayor prioridad para atacar. Si ambos enemigos tienen la misma prioridad, selecciona el más cercano.

Entrada:

enemy1: "EnemyA"

dist1: 20 priority1: 2

Salida:

EnemyB dist1:

enemy2: "EnemyB"

dist2: 15 priority2: 3

```
// Función principal
function seleccionarEnemigo() {
  // Continuarmente leer entradas
  while (true) {
    // Leer los nombres, distancias y prioridades de los enemigos
    let enemy1 = prompt("Ingrese el nombre del primer enemigo:");
    let dist1 = parseFloat(prompt("Ingrese la distancia del primer enemigo:"));
    let priority1 = parseInt(prompt("Ingrese la prioridad del primer enemigo:"));

    let enemy2 = prompt("Ingrese el nombre del segundo enemigo:");
    let dist2 = parseFloat(prompt("Ingrese la distancia del segundo enemigo:"));
    let priority2 = parseInt(prompt("Ingrese la prioridad del segundo enemigo:"));

    // Determinar cuál enemigo atacar
    if (priority1 > priority2) {
      console.log(enemy1);
    } else if (priority2 > priority1) {
      console.log(enemy2);
    } else {
      // Si las prioridades son iguales, seleccionar el más cercano
      if (dist1 < dist2) {
        console.log(enemy1);
      } else {
        console.log(enemy2);
      }
    }

    // Esperar un corto período de tiempo antes de la siguiente lectura
    setTimeout(() => {}, 500); // 500 ms de espera
  }
}

// Ejecutar la función principal
seleccionarEnemigo();
```

Problema V (30 puntos)

Desarrolla un programa de simulación de radar que lea continuamente el nombre, la distancia, la velocidad y la prioridad de ataque de dos enemigos. El programa debe seleccionar el enemigo con mayor prioridad para atacar. Si ambos tienen la misma prioridad, selecciona el más cercano. Si las distancias son iguales, selecciona el enemigo con mayor velocidad.

Entrada:
enemy1: "Enemy1"

Salida:
Enemy2

dist1: 30 priority1: 2
speed1: 10

enemy2: "Enemy2"
dist2: 30 priority2: 2
speed2: 20

```

1 // Función principal
2 function simulacionRadar() {
3     // Continuarmente leer entradas
4     while (true) {
5         // Leer los nombres, distancias, prioridades y velocidades de los enemigos
6         let enemy1 = prompt("Ingrese el nombre del primer enemigo:");
7         let dist1 = parseFloat(prompt("Ingrese la distancia del primer enemigo:"));
8         let priority1 = parseInt(prompt("Ingrese la prioridad del primer enemigo:"));
9         let speed1 = parseFloat(prompt("Ingrese la velocidad del primer enemigo:"));
10
11         let enemy2 = prompt("Ingrese el nombre del segundo enemigo:");
12         let dist2 = parseFloat(prompt("Ingrese la distancia del segundo enemigo:"));
13         let priority2 = parseInt(prompt("Ingrese la prioridad del segundo enemigo:"));
14         let speed2 = parseFloat(prompt("Ingrese la velocidad del segundo enemigo:"));
15
16         // Determinar cuál enemigo atacar
17         let enemigoSeleccionado;
18         if (priority1 > priority2) {
19             enemigoSeleccionado = enemy1;
20         } else if (priority2 > priority1) {
21             enemigoSeleccionado = enemy2;
22         } else {
23             // Si las prioridades son iguales, seleccionar el más cercano
24             if (dist1 < dist2) {
25                 enemigoSeleccionado = enemy1;
26             } else if (dist2 < dist1) {
27                 enemigoSeleccionado = enemy2;
28             } else {
29                 // Si las distancias son iguales, seleccionar el de mayor velocidad
30                 if (speed1 > speed2) {
31                     enemigoSeleccionado = enemy1;
32                 } else {
33                     enemigoSeleccionado = enemy2;
34                 }
35             }
36         }
37     }

```

```

        console.log(enemigoSeleccionado);

        // Esperar un corto período de tiempo antes de la siguiente lectura
        setTimeout(() => {}, 500); // 500 ms de espera
    }
}

// Ejecutar la función principal
simulacionRadar();

```

