

Primer mensaje por consola

The screenshot shows the JavaScript Visualizer 9000 interface. The code editor on the left contains a script with a `setTimeout` and a `Promise`. The Task Queue on the right contains a single task labeled "Microtask 1 (Promise)". The Microtask Queue is empty. The Call Stack is empty. The Event Loop is in the "Evaluate Script" phase, with a description: "Synchronously execute the script as though it were a function body. Run until the Call Stack is empty." The "Run all Microtasks" button is highlighted.

Segundo mensaje por consola

The screenshot shows the JavaScript Visualizer 9000 interface. The code editor on the left shows the script. The Task Queue now contains two tasks, both labeled "anonymous". The Microtask Queue is still empty. The Call Stack is empty. The Event Loop is in the "Run all Microtasks" phase, with a description: "Select the oldest Microtask from the Microtask Queue. Run it until the Call Stack is empty." The "Run all Microtasks" button is highlighted.

Tercer mensaje por consola

The screenshot shows the JavaScript Visualizer 9000 interface. The code editor on the left shows the script. The Task Queue now contains three tasks, all labeled "anonymous". The Microtask Queue is still empty. The Call Stack is empty. The Event Loop is in the "Run all Microtasks" phase, with a description: "Select the oldest Microtask from the Microtask Queue. Run it until the Call Stack is empty." The "Run all Microtasks" button is highlighted.

Cuarto mensaje por consola

The screenshot shows the JavaScript Visualizer 9000 interface. The code editor on the left contains a script with several `setTimeout` and `console.log` statements. The right panel displays the execution state:

- Task Queue:** Three items, all labeled "anonymous".
- Microtask Queue:** One item, labeled "anonymous".
- Call Stack:** One item, labeled "anonymous".
- Event Loop:** A list of steps: "Evaluate Script" (checked), "Run a Task" (checked), and "Run all Microtasks" (selected with a blue circle). Below the list, it says: "Select the oldest Microtask from the Microtask Queue. Run it until the Call Stack is empty."

At the bottom right, there are buttons for "STEP" and "STOP".

Quinto mensaje por consola

The screenshot shows the JavaScript Visualizer 9000 interface. The code editor on the left contains the same script. The right panel displays the execution state:

- Task Queue:** Three items, all labeled "anonymous".
- Microtask Queue:** One item, labeled "anonymous".
- Call Stack:** One item, labeled "anonymous".
- Event Loop:** A list of steps: "Evaluate Script" (checked), "Run a Task" (checked), and "Run all Microtasks" (selected with a blue circle). Below the list, it says: "Select the oldest Microtask from the Microtask Queue. Run it until the Call Stack is empty."

At the bottom right, there are buttons for "STEP" and "STOP".

Sexto mensaje por consola

The screenshot shows the JavaScript Visualizer 9000 interface. The code editor on the left contains the same script. The right panel displays the execution state:

- Task Queue:** Two items, both labeled "anonymous".
- Microtask Queue:** One item, labeled "anonymous".
- Call Stack:** One item, labeled "anonymous".
- Event Loop:** A list of steps: "Evaluate Script" (checked), "Run a Task" (selected with a blue circle), and "Run all Microtasks". Below the list, it says: "Select the oldest Task from the Task Queue. Run it until the Call Stack is empty."

At the bottom right, there are buttons for "STEP" and "STOP".

Séptimo mensaje por consola

The screenshot shows the JavaScript Visualizer 9000 interface. The code editor on the left contains a script with several `setTimeout` and `Promise` calls. The right panel displays the execution state:

- Task Queue:** Contains an `anonymous` task.
- Microtask Queue:** Contains three items: `Microtask 2 (Promise)`, `Microtask 4 (Promise)`, and `Microtask 0 seconds (setTimeout)`.
- Call Stack:** Shows an `anonymous` function.
- Event Loop:** Shows the current step: `Run a Task`, with instructions: "Select the oldest Task from the Task Queue. Run it until the Call Stack is empty."

Octavo mensaje por consola

The screenshot shows the JavaScript Visualizer 9000 interface at a later stage than the previous one. The code editor on the left shows the same script. The right panel displays the updated execution state:

- Task Queue:** Still contains the `anonymous` task.
- Microtask Queue:** Now contains two items: `Microtask 0 seconds (setTimeout)` and `Microtask 1 second (setTimeout)`. The `Microtask 4 (Promise)` and `Microtask 2 (Promise)` have been processed.
- Call Stack:** Still shows the `anonymous` function.
- Event Loop:** Still shows the `Run a Task` step.

- ¿Qué tareas se consideran macrotareas y cuáles son microtareas?

Las macrotareas son los `setTimeout` o las web APIs y las microtareas son las promesas.

- ¿Cómo se relacionan las macrotareas y microtareas con el event loop?

El ciclo de eventos ejecuta primero las actividades síncronas y luego las asíncronas. Las primeras actividades en ejecutar en el entorno asíncrono son las microtareas y luego las macrotareas.

- ¿Qué sucede cuando una microtarea genera una nueva macrotarea dentro de ella?

La microtarea se resuelve al ingresar la macrotarea al task queue. La macrotarea se resolverá una vez se finalice la ejecución de todas las microtareas.

- ¿Cómo se manejan las promesas y los setTimeout en relación con el event loop?

Las promesas son microtareas que se resuelven en el evento loop una vez finaliza la ejecución de las actividades síncronas. Los setTimeout son macrotareas y se resuelven en el event loop una vez son ejecutadas las microtareas.