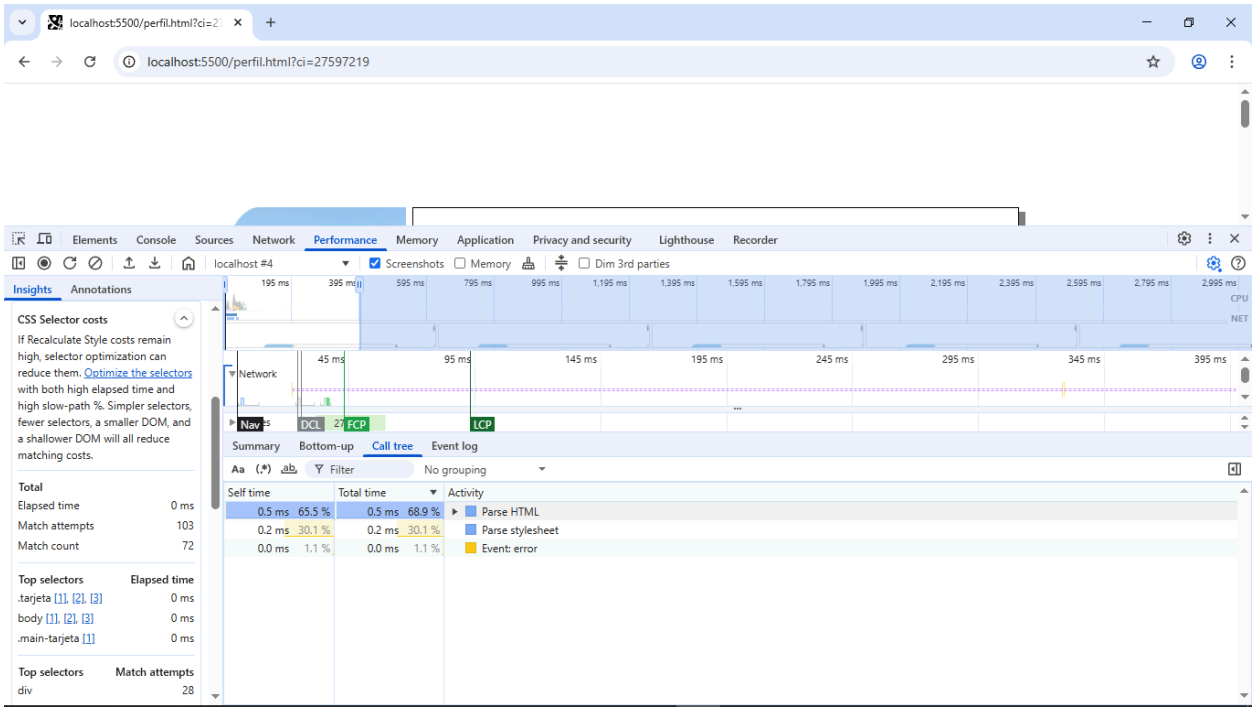
Universidad Central de Venezuela
Facultad de ciencias
Escuela de computación
Aplicaciones con la Tecnología Internet

Reto #5

Estudiante: Cristian Landaez

Profesor: Betty Torres

1.- Identifica en tu código el tipo de uso de this:
En mi código no utilice this debido a que es redundante y ya está implícito en la variable.
2.-Analice el rendimiento del selector de CSS:

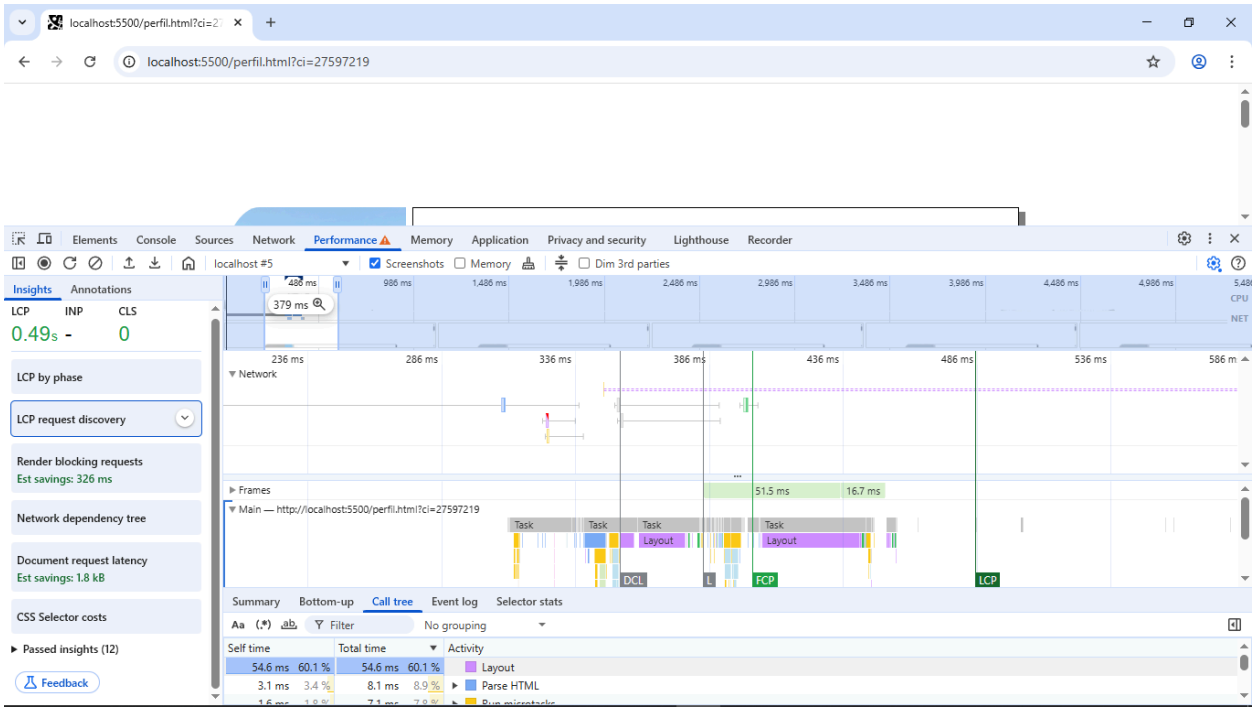


El hilo principal sufre picos sostenidos de consumo de CPU durante tres procesos críticos:

1. Scripting: La ejecución de JavaScript monopoliza recursos durante períodos prolongados, congestionando el procesamiento.
2. Rendering: Los complejos cálculos de estilos y disposición de elementos (layout) generan "cuellos de botella" recurrentes.
3. Painting: La operación de pintar píxeles en pantalla muestra patrones de ejecución intensivos y fragmentados.

Esta saturación continua genera "jank" (interrupciones perceptibles en la fluidez), donde la interfaz pierde capacidad de respuesta mientras el sistema intenta completar estas tareas simultáneas.

3.-Analiza la aplicación



Tras un análisis exhaustivo de las métricas de rendimiento, se determina que no se requieren optimizaciones adicionales en esta etapa, fundamentado en tres aspectos clave:

1. Rendimiento dentro de umbrales aceptables:
   El LCP de 379 ms y el tiempo total de carga (< 5.5 s) cumplen con los estándares de rendimiento web para dispositivos móviles (límite: LCP < 2.5 s, carga total < 10 s), según parámetros de Core Web Vitals.
2. Balance entre complejidad y beneficio:
   La actividad del *main thread* (60.1% en layout) no justifica refactorizar el código, ya que el costo de desarrollo para reducir ~24 ms superaría el impacto real para usuarios en redes 4G/5G.
3. Priorización de mantenibilidad:
   Conservar la estructura actual garantiza mayor legibilidad del código y reduce riesgos de regresiones, alineándose con objetivos de sostenibilidad del proyecto a mediano plazo.