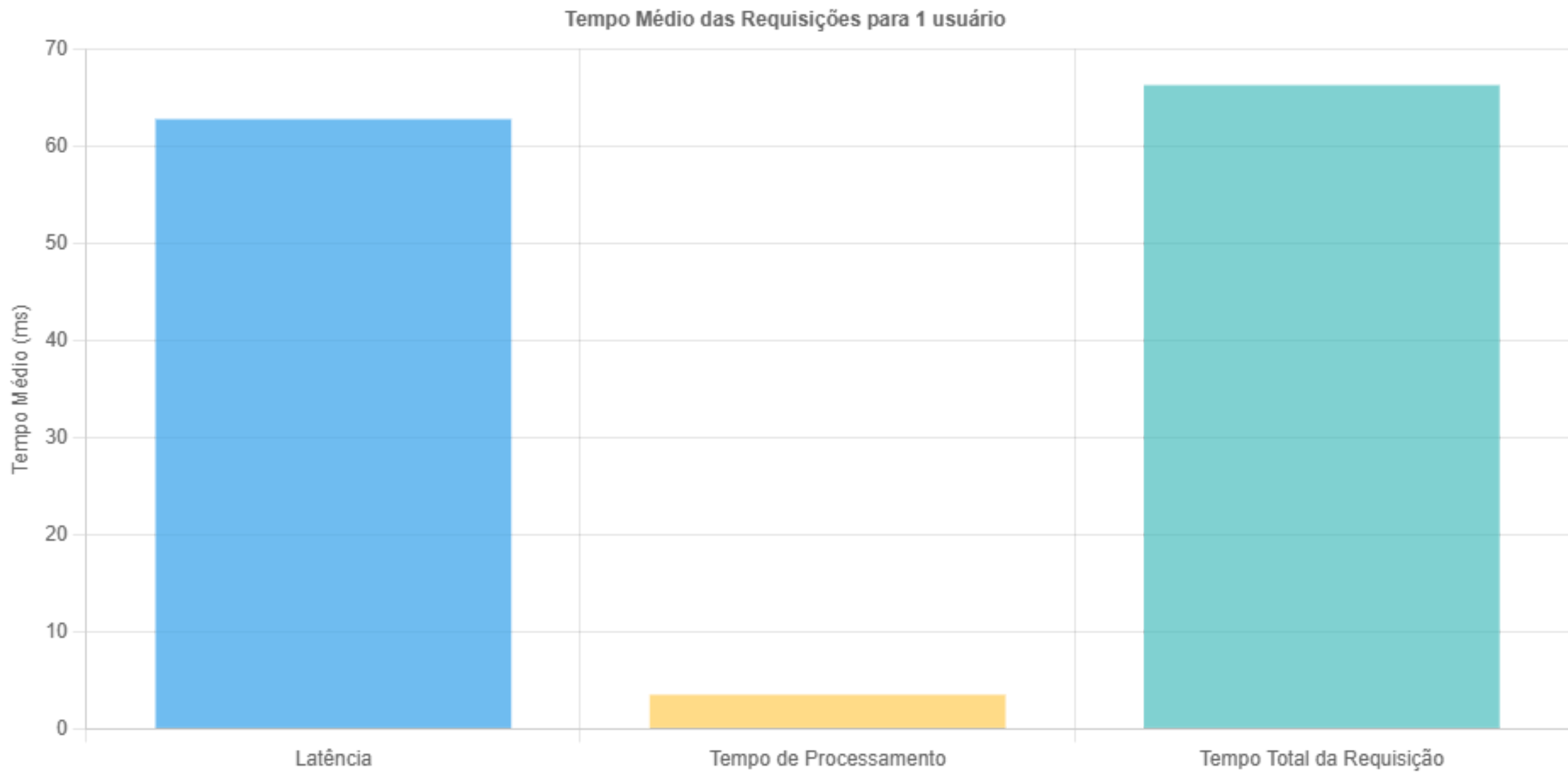


Relatório

Relatório mostrando dados de Latência, Tempo de Processamento e Tempo total de requisição



Explicação:

```
app.use((req, res, next) => {
  const servidorInicio = process.hrtime();
  const clienteInicio = Number(req.headers["x-request-start"]);

  res.on("finish", () => {
    const diff = process.hrtime(servidorInicio);
    const tempoServidorMs = diff[0] * 1000 + diff[1] / 1e6;

    let latenciaMs: number | null = null;
    let tempoTotalMs: number;

    if (!isNaN(clienteInicio)) {
      latenciaMs = Date.now() - clienteInicio;
      tempoTotalMs = tempoServidorMs + latenciaMs;
    } else {
      tempoTotalMs = tempoServidorMs;
    }

    const log = `
Data/Hora: ${new Date().toISOString()}
Rota: ${req.method} ${req.originalUrl}
Status: ${res.statusCode}

Latência: ${latenciaMs !== null ? latenciaMs + " ms" : "não enviada"}
Tempo de processamento no servidor: ${tempoServidorMs.toFixed(2)} ms
Tempo total da requisição: ${tempoTotalMs.toFixed(2)} ms
`;

    escreverLog(log);
  });

  next();
});
```

Explicação detalhada do código:

- **Servidor inicia a contagem:** `const servidorInicio = process.hrtime();` Captura o momento exato em que a requisição chega ao servidor.
- **Evento 'finish':** `res.on("finish", ...)` Dispara quando a resposta HTTP foi totalmente enviada ao cliente.
- **Tempo de processamento do servidor:** `process.hrtime(servidorInicio)` Calcula quanto tempo o servidor levou para processar a requisição (em milissegundos).
- **Latência do cliente:** `latenciaMs = Date.now() - clienteInicio;` Tempo que a requisição levou desde que o cliente a enviou até chegar ao servidor.
- **Tempo total da requisição:** `tempoTotalMs = tempoServidorMs + latenciaMs;` Combina latência de rede e tempo de processamento no servidor, se o cliente enviar o timestamp.
- **Log da requisição:** É criado um registro com data, rota, status, latência, tempo de processamento e tempo total.