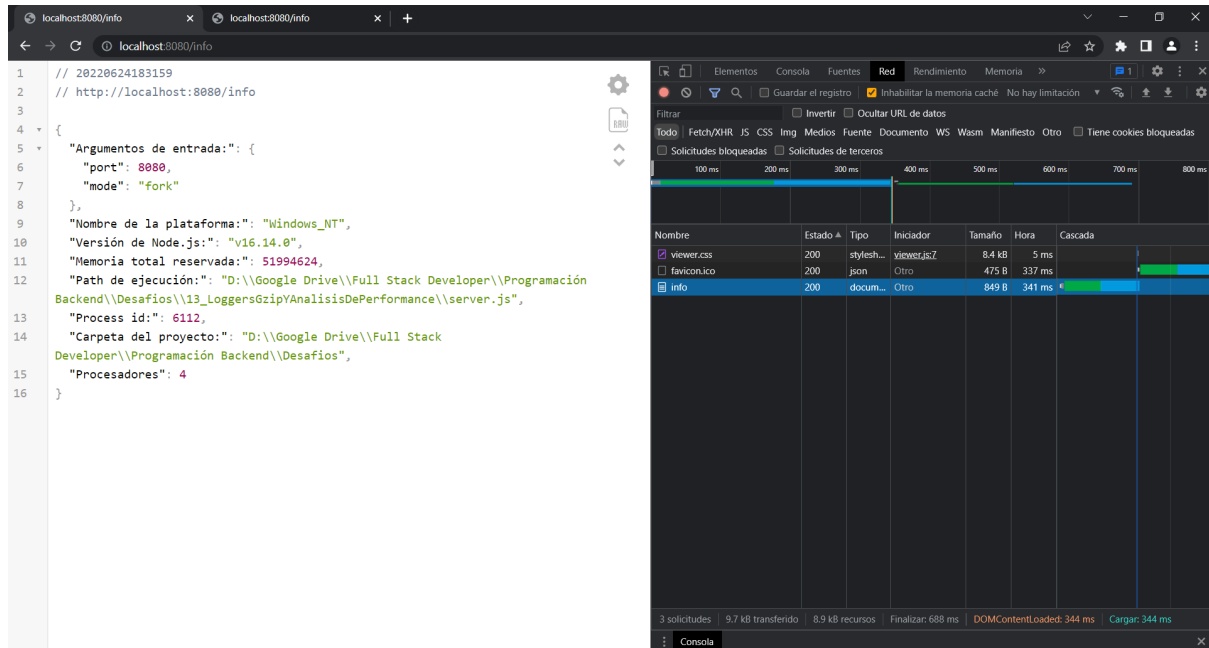
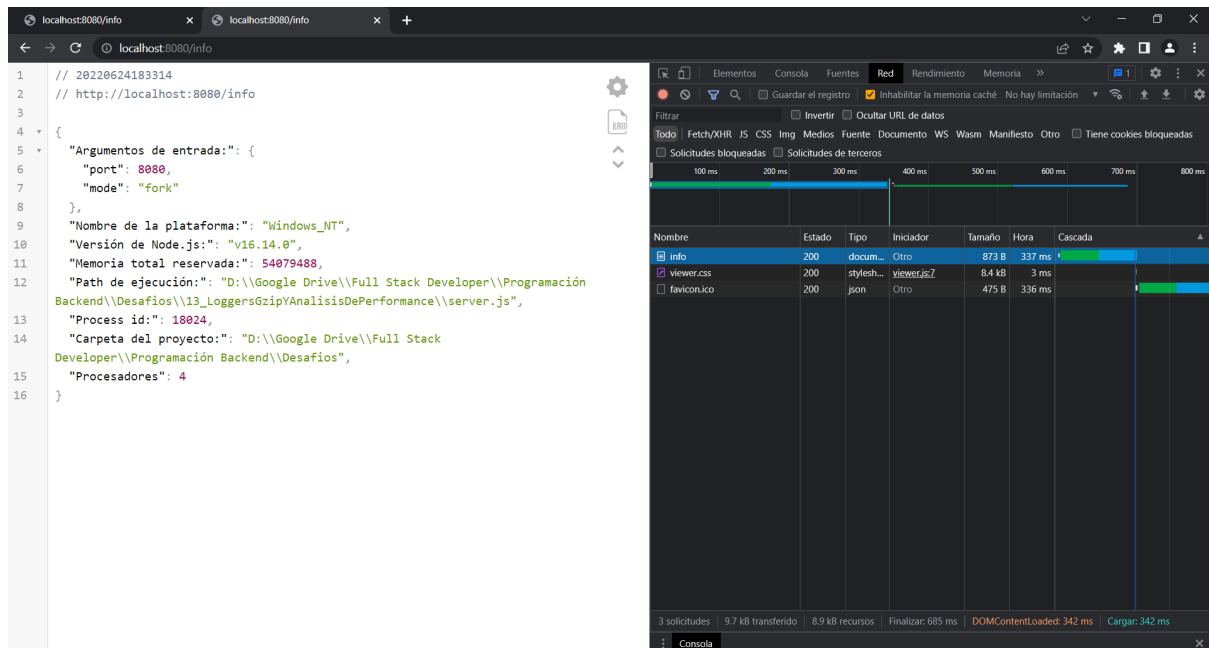


# ANÁLISIS DE PERFORMANCE

- Ruta /info **sin compression**:



- Ruta /info **con compression**:



- Resultado **Artillery** sobre ruta /info:

```

111 All WUs finished. Total time: 49 seconds
112
113 Summary report @ 23:02:14(-0300)
114
115 -----
116 http.codes.200: ..... 1000
117 http.request_rate: ..... 13/sec
118 http.requests: ..... 1000
119 http.response_time:
120 min: ..... 2
121 max: ..... 6492
122 median: ..... 871.5
123 p95: ..... 3197.8
124 p99: ..... 4867
125 http.responses: ..... 1000
126 users.completed: ..... 50
127 users.created: ..... 50
128 users.created_by_name.0: ..... 50
129 users.failed: ..... 0
130 users.session_length:
131 min: ..... 45831.3
132 max: ..... 47586.2
133 median: ..... 46644.4
134 p95: ..... 47586.7
135 p99: ..... 47586.7
136
137
  
```

Terminal output (bottom):

```

{"level":"warn","message":"ruta /favicon.ico metodo GET no autorizada"}
{"level":"info","message":"ruta /info metodo GET autorizada"}
{"level":"warn","message":"ruta /favicon.ico metodo GET no autorizada"}
  
```

- Resultado **Artillery** sobre ruta /info-console:

```

111 All WUs finished. Total time: 50 seconds
112
113 Summary report @ 23:04:08(-0300)
114
115 -----
116 http.codes.200: ..... 1000
117 http.request_rate: ..... 13/sec
118 http.requests: ..... 1000
119 http.response_time:
120 min: ..... 3
121 max: ..... 6528
122 median: ..... 907
123 p95: ..... 3262.4
124 p99: ..... 4965.3
125 http.responses: ..... 1000
126 users.completed: ..... 50
127 users.created: ..... 50
128 users.created_by_name.0: ..... 50
129 users.failed: ..... 0
130 users.session_length:
131 min: ..... 46963.2
132 max: ..... 48665.1
133 median: ..... 47586.7
134 p95: ..... 48548.1
135 p99: ..... 48548.1
136
137
138
  
```

Terminal output (bottom):

```

{"level":"warn","message":"ruta /favicon.ico metodo GET no autorizada"}
{"level":"info","message":"ruta /info metodo GET autorizada"}
{"level":"warn","message":"ruta /favicon.ico metodo GET no autorizada"}
  
```

- Resultado **profiling con -prof** sobre ruta /info:

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer Pane (Left):** Displays the file structure of a project. The file `result_prof-noblog.txt` is selected and highlighted.
- Main Editor:** Displays the content of `result_prof-noblog.txt`.
  - [++]:** A table showing ticks, total, nonlib, and name for various components.
 

	ticks	total	nonlib	name
[Summary]:				
	222	0.5%	98.2%	JavaScript
	0	0.0%	0.0%	C++
	98	0.2%	43.4%	GC
	47295	99.5%		Shared libraries
	4	0.0%		Unaccounted
  - [C++ entry points]:** A table showing ticks, cpp, total, and name.
 

	ticks	cpp	total	name
[Bottom up (heavy) profile]:				
Note: percentage shows a share of a particular caller in the total amount of its parent calls.				
callers occupying less than 1.0% are not shown.				
	ticks	parent	name	
	43824	92.2%	C:\Windows\SYSTEM32\ntdll.dll	
	3461	7.3%	C:\Program Files\nodejs\node.exe	
	2278	65.8%	C:\Program Files\nodejs\node.exe	
	305	13.4%	Function: ~compilofunction node:vm:308:25	
	294	96.4%	Function: ~wrapSafe node:internal/modules/cjs/loader:1017:18	
	272	92.5%	Function: ~Module._compile node:internal/modules/cjs/loader:1057:37	
	269	98.0%	Function: ~Module._extensions..js node:internal/modules/cjs/loader:1112:37	
	3	1.1%	LazyCompile: ~Module._extensions..js node:internal/modules/cjs/loader:1112:37	
	22	7.5%	LazyCompile: ~Module._compile node:internal/modules/cjs/loader:1057:37	
- Terminal Pane (Bottom):** Shows the output of the application, including warnings and information messages.
 

```
{
  "level": "warn",
  "message": "ruta /favicon.ico metodo GET no autorizada"
}
{
  "level": "info",
  "message": "ruta /info metodo GET autorizada"
}
{
  "level": "warn",
  "message": "ruta /favicon.ico metodo GET no autorizada"
}
```

- Resultado **profiling con -prof** sobre ruta /info-console:

Visual Studio Code interface showing a performance profile of a Node.js application. The Explorer pane on the left shows the file structure of a project named '13\_loggersGzipVAnalisisDePerformance'. The main editor displays the file 'result\_curl\_prof-bloq(console).txt', which contains a performance profile. The profile includes a summary table, entry points, and a bottom-up (heavy) profile. The bottom-up profile shows the call stack for various functions, including 'handleWriteReq', 'writeGeneric', 'Socket\_writeGeneric', 'Socket\_write', 'compileFunction', 'wrapSafe', and 'Module\_compile'. The bottom status bar shows the current file is 'main.js' and the editor is in 'Scanning...' mode.

**File Explorer:**

- 13\_loggersGzipVAnalisisDePerformance
  - controllers
  - db
  - logs
  - middlewares
  - profiling
    - 7420.0x
    - bloq-v8.log
    - nobloq-v8.log
    - result\_autocannon\_console...
    - result\_autocannon.png
    - result\_curl\_console.txt
    - result\_curl\_prof-bloq(console).txt**
    - result\_curl\_prof-nobloq.txt
    - result\_curl.txt
    - result\_inspect\_console.png
    - result\_inspect\_console.txt
    - result\_inspect.png
    - result\_inspect.txt
  - public
  - routes
  - views
  - config.js
  - server.js
  - node\_modules
  - .env
  - .gitignore

**Performance Profile Content:**

```
[C++]:
ticks total nonlib name

[Summary]:
ticks total nonlib name
192 0.3% 99.0% JavaScript
0 0.0% 0.0% C++
95 0.1% 49.0% GC
66084 99.7% shared libraries
2 0.0% Unaccounted

[C++ entry points]:
ticks cpp total name

[Bottom up (heavy) profile]:
Note: percentage shows a share of a particular caller in the total
amount of its parent calls.
callers occupying less than 1.0% are not shown.

ticks parent name
61693 93.2% C:\Windows\SYSTEM32\ntdll.dll

4296 6.5% C:\Program Files\nodejs\node.exe
3130 72.0% C:\Program Files\nodejs\node.exe
Function: *handleWriteReq node:internal/stream_base_commons:45:24
962 100.0% Function: *writeGeneric node:internal/stream_base_commons:147:22
962 100.0% Function: *Socket_writeGeneric node:net:769:42
962 100.0% Function: *Socket_write node:net:806:35
295 9.4% Function: *compileFunction node:vm:308:25
288 97.6% Function: *wrapSafe node:internal/modules/cjs/loader:1017:18
276 95.8% Function: *Module_compile node:internal/modules/cjs/loader:1057:37
```

**Terminal:**

```
{
  "level": "warn",
  "message": "ruta /favicon.ico metodo GET no autorizada"
}, {
  "level": "info",
  "message": "ruta /info metodo GET autorizada"
}, {
  "level": "warn",
  "message": "ruta /favicon.ico metodo GET no autorizada"
}
```

- Resultado **Autocannon** sobre ruta /info:

```

herre@DESKTOP-LOUDG4Q MINGW64 ~/Desktop/Desafios (main)
$ autocannon -d 20 -c 100 "http://localhost:8080/info"
Running 20s test @ http://localhost:8080/info
100 connections
  
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	60 ms	5720 ms	9413 ms	9740 ms	5577.34 ms	2377.52 ms	9779 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	1	1	11	27	12.7	7.43	1
Bytes/Sec	794 B	794 B	8.78 kB	21.5 kB	16.1 kB	5.91 kB	794 B

Req/Bytes counts sampled once per second.  
 # of samples: 20  
 377 requests in 20.1s, 202 kB read  
 23 errors (23 timeouts)

```

herre@DESKTOP-LOUDG4Q MINGW64 ~/Desktop/Desafios (main)
$
  
```

- Resultado **Autocannon** sobre ruta /info-console:

```

herre@DESKTOP-LOUDG4Q MINGW64 ~/Desktop/Desafios (main)
$ autocannon -d 20 -c 100 "http://localhost:8080/info-console"
Running 20s test @ http://localhost:8080/info-console
100 connections
  
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	344 ms	2060 ms	3393 ms	3425 ms	2110.61 ms	576.5 ms	3482 ms

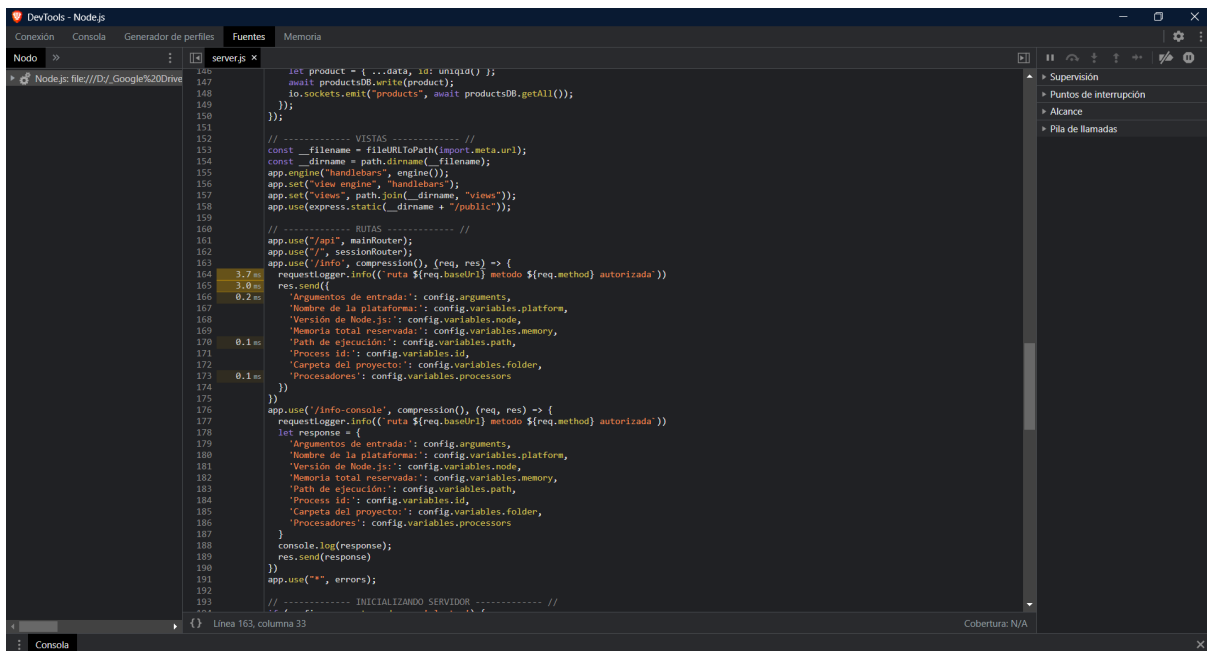
Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	27	27	44	98	46.2	17.04	27
Bytes/Sec	21.5 kB	21.5 kB	35 kB	78.1 kB	36.8 kB	13.6 kB	21.5 kB

Req/Bytes counts sampled once per second.  
 # of samples: 20  
 1k requests in 20.09s, 736 kB read

```

herre@DESKTOP-LOUDG4Q MINGW64 ~/Desktop/Desafios (main)
$
  
```

- Resultado **profiling con -inspect** sobre ruta /info:



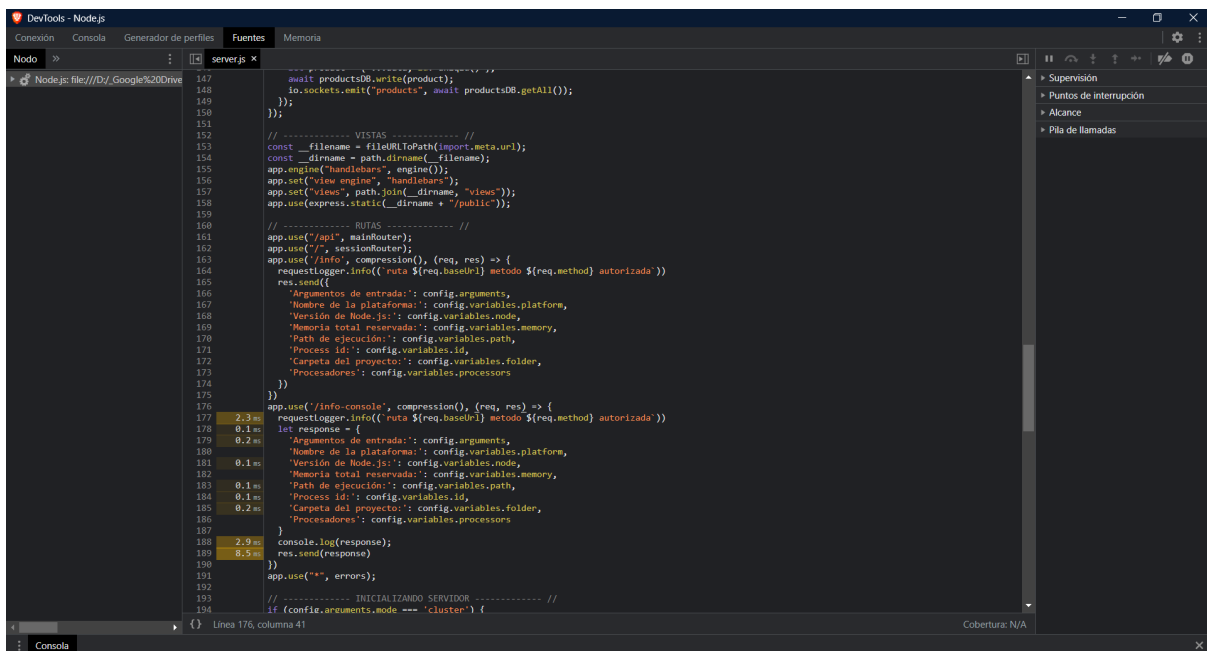
The screenshot shows the VS Code DevTools interface with the 'Fuentes' (Sources) tab selected. The file 'server.js' is open, and the 'Console' tab at the bottom shows the output of the application. The code in 'server.js' includes a route for '/info' that logs request information and sends a response. The profiling data in the console shows the following:

```
146 let product = { ...data, id: uniqueId() };
147 await productsDB.write(product);
148 io.sockets.emit("products", await productsDB.getAll());
149 });
150
151 // ----- VISTAS ----- //
152 const __filename = fileURLToPath(import.meta.url);
153 const __dirname = path.dirname(__filename);
154 app.engine("handlebars", engine());
155 app.set("view engine", "handlebars");
156 app.set("views", path.join(__dirname, "views"));
157 app.use(express.static(__dirname + "/public"));
158
159 // ----- RUTAS ----- //
160 app.use("/api", mainRouter);
161 app.use("/", sessionRouter);
162 app.use("/info", compression(), (req, res) => {
163   requestLogger.info(`ruta ${req.baseUrl} metodo ${req.method} autorizada`);
164   res.send({
165     'Argumentos de entrada': config.arguments,
166     'Nombre de la plataforma': config.variables.platform,
167     'Versión de Node.js': config.variables.node,
168     'Memoria total reservada': config.variables.memory,
169     'Path de ejecución': config.variables.path,
170     'Process id': config.variables.id,
171     'Carpeta del proyecto': config.variables.folder,
172     'Procesadores': config.variables.processors
173   });
174 });
175
176 app.use("/info-console", compression(), (req, res) => {
177   requestLogger.info(`ruta ${req.baseUrl} metodo ${req.method} autorizada`);
178   let response = {
179     'Argumentos de entrada': config.arguments,
180     'Nombre de la plataforma': config.variables.platform,
181     'Versión de Node.js': config.variables.node,
182     'Memoria total reservada': config.variables.memory,
183     'Path de ejecución': config.variables.path,
184     'Process id': config.variables.id,
185     'Carpeta del proyecto': config.variables.folder,
186     'Procesadores': config.variables.processors
187   };
188   console.log(response);
189   res.send(response);
190 });
191 app.use("/*", errors);
192
193 // ----- INICIALIZANDO SERVIDOR ----- //
```

The console output shows the following:

```
146 let product = { ...data, id: uniqueId() };
147 await productsDB.write(product);
148 io.sockets.emit("products", await productsDB.getAll());
149 });
150
151 // ----- VISTAS ----- //
152 const __filename = fileURLToPath(import.meta.url);
153 const __dirname = path.dirname(__filename);
154 app.engine("handlebars", engine());
155 app.set("view engine", "handlebars");
156 app.set("views", path.join(__dirname, "views"));
157 app.use(express.static(__dirname + "/public"));
158
159 // ----- RUTAS ----- //
160 app.use("/api", mainRouter);
161 app.use("/", sessionRouter);
162 app.use("/info", compression(), (req, res) => {
163   requestLogger.info(`ruta ${req.baseUrl} metodo ${req.method} autorizada`);
164   res.send({
165     'Argumentos de entrada': config.arguments,
166     'Nombre de la plataforma': config.variables.platform,
167     'Versión de Node.js': config.variables.node,
168     'Memoria total reservada': config.variables.memory,
169     'Path de ejecución': config.variables.path,
170     'Process id': config.variables.id,
171     'Carpeta del proyecto': config.variables.folder,
172     'Procesadores': config.variables.processors
173   });
174 });
175
176 app.use("/info-console", compression(), (req, res) => {
177   requestLogger.info(`ruta ${req.baseUrl} metodo ${req.method} autorizada`);
178   let response = {
179     'Argumentos de entrada': config.arguments,
180     'Nombre de la plataforma': config.variables.platform,
181     'Versión de Node.js': config.variables.node,
182     'Memoria total reservada': config.variables.memory,
183     'Path de ejecución': config.variables.path,
184     'Process id': config.variables.id,
185     'Carpeta del proyecto': config.variables.folder,
186     'Procesadores': config.variables.processors
187   };
188   console.log(response);
189   res.send(response);
190 });
191 app.use("/*", errors);
192
193 // ----- INICIALIZANDO SERVIDOR ----- //
```

- Resultado **profiling con -inspect** sobre ruta /info-console:



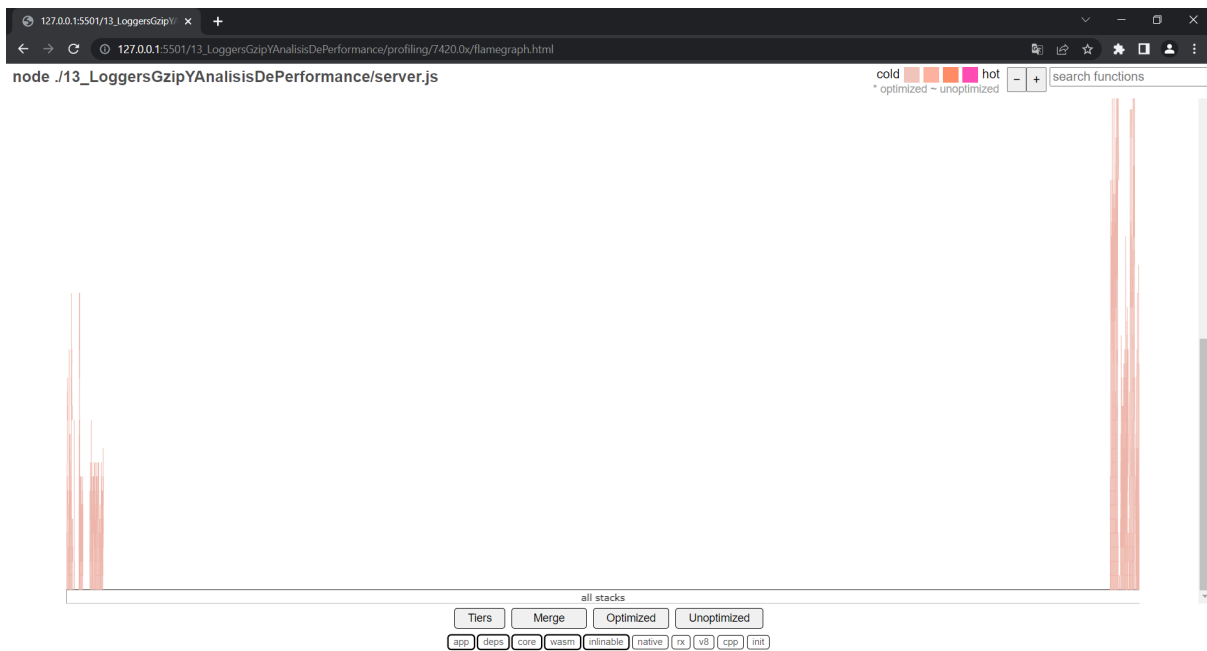
The screenshot shows the VS Code DevTools interface with the 'Fuentes' (Sources) tab selected. The file 'server.js' is open, and the 'Console' tab at the bottom shows the output of the application. The code in 'server.js' includes a route for '/info-console' that logs request information and sends a response. The profiling data in the console shows the following:

```
147 await productsDB.write(product);
148 io.sockets.emit("products", await productsDB.getAll());
149 });
150
151 // ----- VISTAS ----- //
152 const __filename = fileURLToPath(import.meta.url);
153 const __dirname = path.dirname(__filename);
154 app.engine("handlebars", engine());
155 app.set("view engine", "handlebars");
156 app.set("views", path.join(__dirname, "views"));
157 app.use(express.static(__dirname + "/public"));
158
159 // ----- RUTAS ----- //
160 app.use("/api", mainRouter);
161 app.use("/", sessionRouter);
162 app.use("/info", compression(), (req, res) => {
163   requestLogger.info(`ruta ${req.baseUrl} metodo ${req.method} autorizada`);
164   res.send({
165     'Argumentos de entrada': config.arguments,
166     'Nombre de la plataforma': config.variables.platform,
167     'Versión de Node.js': config.variables.node,
168     'Memoria total reservada': config.variables.memory,
169     'Path de ejecución': config.variables.path,
170     'Process id': config.variables.id,
171     'Carpeta del proyecto': config.variables.folder,
172     'Procesadores': config.variables.processors
173   });
174 });
175
176 app.use("/info-console", compression(), (req, res) => {
177   requestLogger.info(`ruta ${req.baseUrl} metodo ${req.method} autorizada`);
178   let response = {
179     'Argumentos de entrada': config.arguments,
180     'Nombre de la plataforma': config.variables.platform,
181     'Versión de Node.js': config.variables.node,
182     'Memoria total reservada': config.variables.memory,
183     'Path de ejecución': config.variables.path,
184     'Process id': config.variables.id,
185     'Carpeta del proyecto': config.variables.folder,
186     'Procesadores': config.variables.processors
187   };
188   console.log(response);
189   res.send(response);
190 });
191 app.use("/*", errors);
192
193 // ----- INICIALIZANDO SERVIDOR ----- //
```

The console output shows the following:

```
147 await productsDB.write(product);
148 io.sockets.emit("products", await productsDB.getAll());
149 });
150
151 // ----- VISTAS ----- //
152 const __filename = fileURLToPath(import.meta.url);
153 const __dirname = path.dirname(__filename);
154 app.engine("handlebars", engine());
155 app.set("view engine", "handlebars");
156 app.set("views", path.join(__dirname, "views"));
157 app.use(express.static(__dirname + "/public"));
158
159 // ----- RUTAS ----- //
160 app.use("/api", mainRouter);
161 app.use("/", sessionRouter);
162 app.use("/info", compression(), (req, res) => {
163   requestLogger.info(`ruta ${req.baseUrl} metodo ${req.method} autorizada`);
164   res.send({
165     'Argumentos de entrada': config.arguments,
166     'Nombre de la plataforma': config.variables.platform,
167     'Versión de Node.js': config.variables.node,
168     'Memoria total reservada': config.variables.memory,
169     'Path de ejecución': config.variables.path,
170     'Process id': config.variables.id,
171     'Carpeta del proyecto': config.variables.folder,
172     'Procesadores': config.variables.processors
173   });
174 });
175
176 app.use("/info-console", compression(), (req, res) => {
177   requestLogger.info(`ruta ${req.baseUrl} metodo ${req.method} autorizada`);
178   let response = {
179     'Argumentos de entrada': config.arguments,
180     'Nombre de la plataforma': config.variables.platform,
181     'Versión de Node.js': config.variables.node,
182     'Memoria total reservada': config.variables.memory,
183     'Path de ejecución': config.variables.path,
184     'Process id': config.variables.id,
185     'Carpeta del proyecto': config.variables.folder,
186     'Procesadores': config.variables.processors
187   };
188   console.log(response);
189   res.send(response);
190 });
191 app.use("/*", errors);
192
193 // ----- INICIALIZANDO SERVIDOR ----- //
```

- Resultado **diagrama de flama con 0x** sobre rutas /info e /info-console:



## CONCLUSIÓN:

Al tratarse de tan pocos datos la compresión no muestra ninguna mejoría en la ruta /info. Por otro lado, al comparar las rutas /info e /info-console (esta última muestra por consola los resultados antes de enviarlos al navegador) los resultados de la prueba de artillery muestran que el comportamiento es similar en ambos casos, no significando la ejecución del `console.log` un problema en este ejemplo. En cuanto al perfilamiento se observan más ticks de librerías en la ruta /info-console, sin embargo, los ticks de JavaScript resultaron menores.