

# proyecto-final-ecommerce

Análisis de performance

# El ejercicio

## Objetivo

Analizaremos el comportamiento, en términos de performance, de nuestra aplicación node a través de diferentes herramientas de evaluación. Se busca medir el impacto de levantar la aplicación en modo fork o modo cluster

## Plan de ejecución

Levantaremos la aplicación en modo *Fork* y utilizaremos al endpoint `/producto/listar/` con una prueba de carga de 3 minutos utilizando la herramienta **artillery**. Al finalizar repetiremos la misma prueba en modo *CLUSTER*

## Informe

Se procesará la información generada por las pruebas y se expondrán las conclusiones de las mismas.



# Artillery

# Pasos

## 1- Aplicación

**npm run start:prof**

Inicia el server en modo  
FORK o CLUSTER

## 2- Artillery

**Npm run test:artillery**

Inicia la prueba con 120  
segundos y rampup de  
60 segundos hasta  
alcanzar los 50 hilos

## 3- Node process

**Node --prof-process**

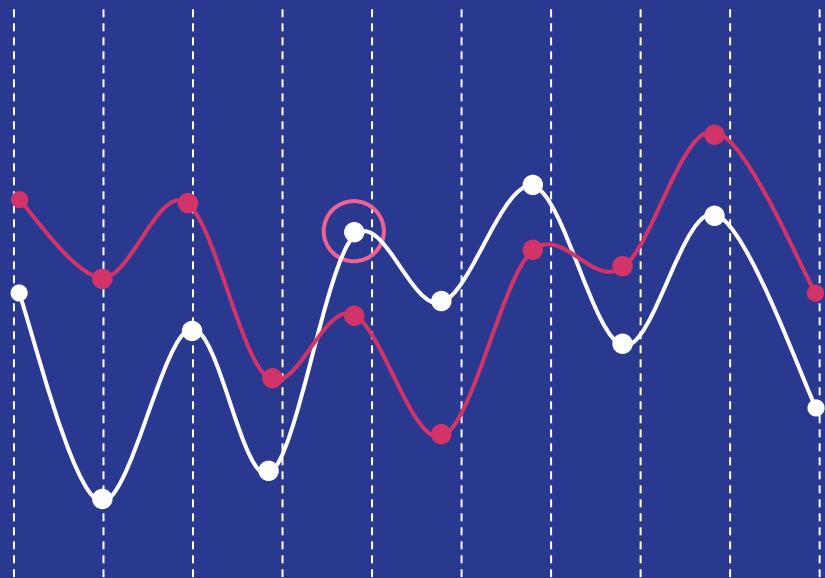
**<<file>>v8.log >**

**<<file>>.txt**

Procesa el reporte

# Reportes

# Server - Fork



```
All virtual users finished
Summary report @ 22:32:13(-0300) 2021-11-08
  Scenarios launched: 3638
  Scenarios completed: 3638
  Requests completed: 3638
  Mean response/sec: 20.12
  Response time (msec):
    min: 158
    max: 2835
    median: 178
    p95: 1634.2
    p99: 2240.6
  Scenario counts:
    Listar: 3638 (100%)
  Codes:
    200: 3638
```

## [Summary]:

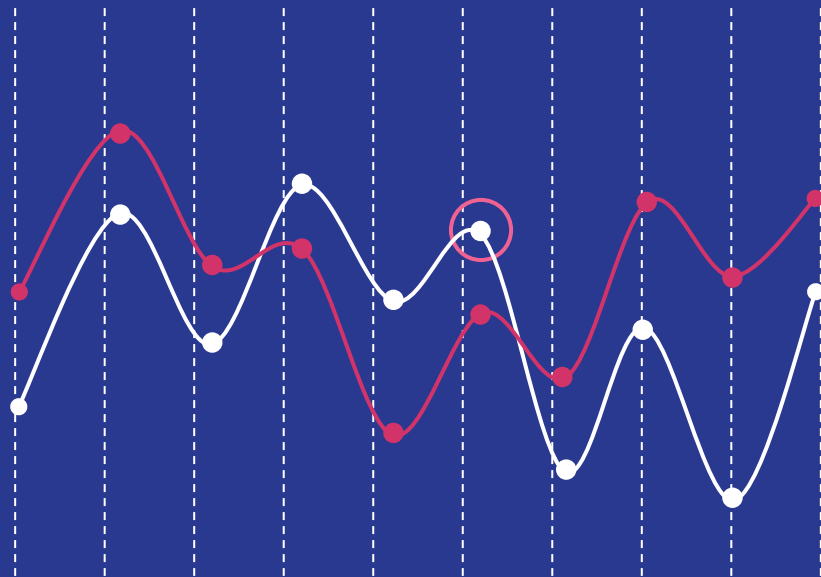
ticks	total	nonlib	name
772	4.1%	4.1%	JavaScript
18001	94.8%	95.7%	C++
308	1.6%	1.6%	GC
174	0.9%		Shared libraries
40	0.2%		Unaccounted

All virtual users finished  
Summary report @ 22:45:45(-0300) 2021-11-08  
Scenarios launched: 3680  
Scenarios completed: 3680  
Requests completed: 3680  
Mean response/sec: 20.35  
Response time (msec):  
  min: 150  
  max: 1958  
  median: 173  
  p95: 353  
  p99: 1154.8  
Scenario counts:  
  Listar: 3680 (100%)  
Codes:  
  200: 3680

[Summary]:

ticks	total	nonlib	name
103	0.8%	0.8%	JavaScript
12627	98.6%	99.1%	C++
253	2.0%	2.0%	GC
69	0.5%		Shared libraries
11	0.1%		Unaccounted

# Server - Cluster



# Node -Profiler ( Fork vs Cluster)

Ticks		TOTAL		nonlib		Name
772	103*	4.1%	0.8%*	4.1%	0.8%*	Javascript
18001	12627*	94.8%	98.6%*	95.7%	99.1%*	C++
308	253*	1.6%	2.0%*	1.6%	2.0%*	GC
174	69*	0.9%	0.5%*	-	-	Shared Libraries
40	11*	0.2%	0.1%*	-	-	Unaccounted

Fork

Cluster





# Informe

# ecommerce en modo fork y cluster

El profiler de node registró **772 Ticks (ciclos)** consumidos para **javascript** con la ejecución **en modo fork** mientras que el modo **cluster** registró **103 Ticks (87% inferior) Shared libraries**,

**C++** registra un menor consumo en ciclos para el **modo cluster** del **30% (12627)** con respecto al **modo fork (18001)**

**Tambien se registran menor cantidad de ciclos para GC, Shared Libraries y Unaccounted** con reducciones del **18% 60,4% y 72,5%** respectivamente

El reporte de Artillery registra una mejora del **50%** en el **%99** al switchear al **modo cluster**, bajando de **2200ms** a **1100ms** en las respuesta, a su vez la media registra una mejora de **178ms** en **mndo fork** a **173ms** en **modo cluster**

*Se concluye que la aplicación corriendo en modo cluster presenta una mejora Notable en tiempos de respuesta como procesamiento.*

