Entrega Nro. 3 – Proyecto Final

**Rutas sin Front-End:**

*Productos:*

GET: <http://localhost:8080/api/productos/:num>

POST: <http://localhost:8080/api/productos>

PUT: <http://localhost:8080/api/productos/:num>

DELETE: <http://localhost:8080/api/productos/:num>

*Carrito:*

GET: <http://localhost:8080/api/carrito/:num>

GET: <http://localhost:8080/api/carrito/:num/productos>

POST: <http://localhost:8080/api/carrito/>

POST: <http://localhost:8080/api/carrito/:num/productos/:num>

POST: <http://localhost:8080/api/carrito/:num/productos/:num/:id_prod>

DELETE: <http://localhost:8080/api/carrito/:num>

DELETE: <http://localhost:8080/api/carrito/:num/productos/:num/:id_prod>

**Proyecto en Railway:**

<https://railwayentrega3-tp-production.up.railway.app/>

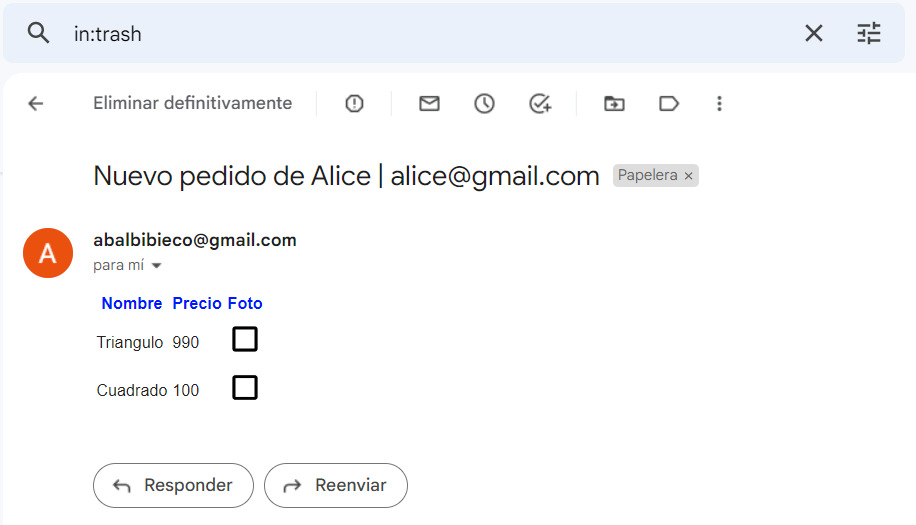
**Proyecto en Railway:**

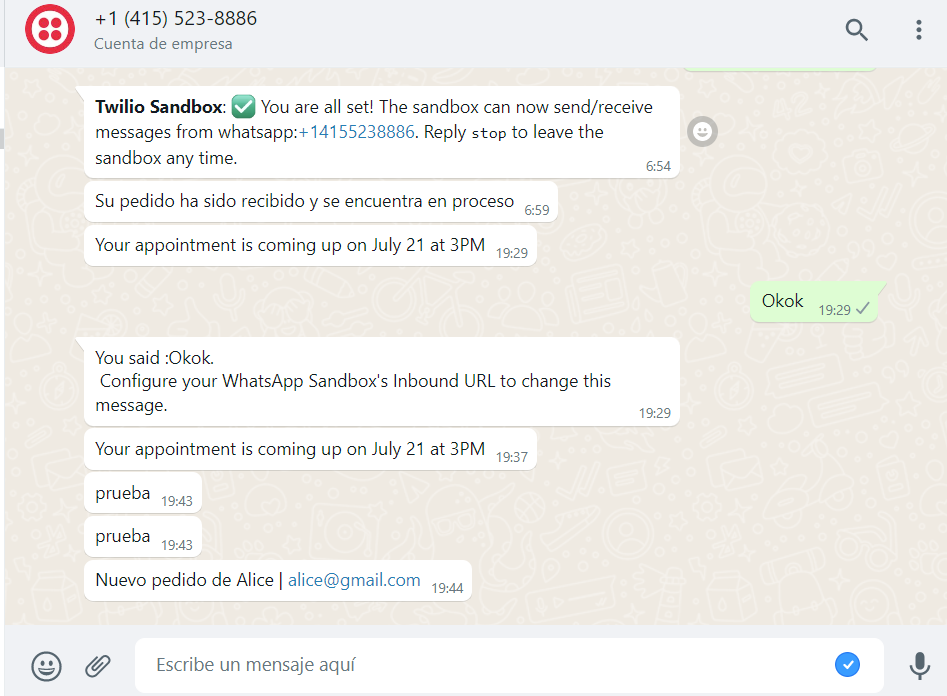
Dentro del proyecto se encontrara el archivo “.env.txt” dado que no subo los archivos .env a Github. A continuación, detallo los mismos para tenerlos a mano:

* **MAKEYTWILIO** = 'ACa2630de2f76186abd32993d9694c367b'
* **MAPASSTWILIO** = '425f76da26b017fd077d6304ee15043a'
* **MAPASSNODEMAILER** = 'fxnspuzlvvcpifbs'
* **MAURLMONGO** = "mongodb+srv://axelken93:Racing200793@backend32190.e4b0dmz.mongodb.net/ecommerce?retryWrites=true&w=majority"

**Twilio + Nodemailer:**

Se hicieron las pruebas con destino de los correos y mensajes hacia mis buzones personales. Adjunto imágenes de WhatsApp y Gmail:





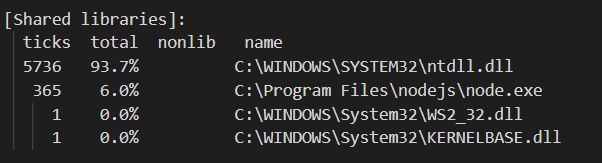
También recibí los SMS desde Twilio pero no tengo las imágenes para adjuntarlas.

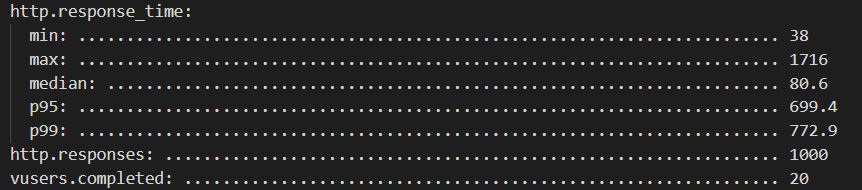
**Prueba de Performance:**

Se realizo una prueba de performance en modo local CON y SIN modo Cluster, utilizando la herramienta Artillery en el endpoint del listado de productos [http://localhost:8080/api/productos](http://localhost:8080/api/productos/:num)

*Sin Cluster:*

* node --prof index.js
* artillery quick --count 20 -n 50 "http://localhost:8080/api/productos" > result\_bloq.txt
* node --prof-process bloq-v8.log > result\_prof-bloq.txt

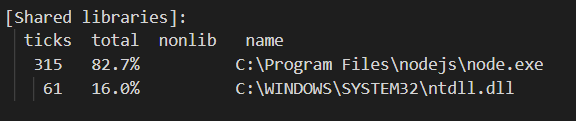


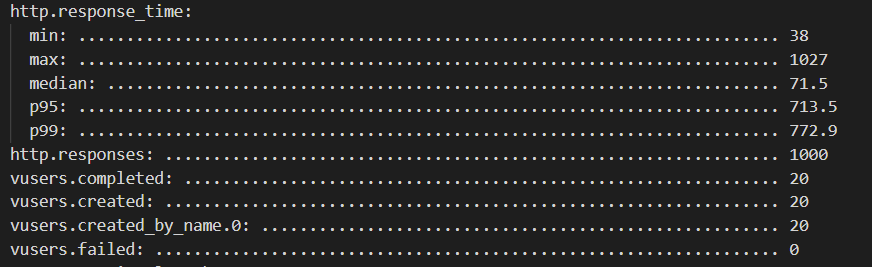


*Con Cluster:*

* node --prof index.js
* artillery quick --count 20 -n 50 "http://localhost:8080/api/productos" > result\_bloq.txt
* node --prof-process bloq-v8.log > result\_prof-bloq.txt

Cada cpu hizo 381 ticks en promedio. En total 3429. Ejemplo de una:





**Conclusiones:**

Con Cluster tuvo un 40% menos de ticks. El tiempo de respuesta mínimo es similar, pero el máximo con cluster es un 40% mas rápido. Se nota una mejoría en la performance utilizando cluster.