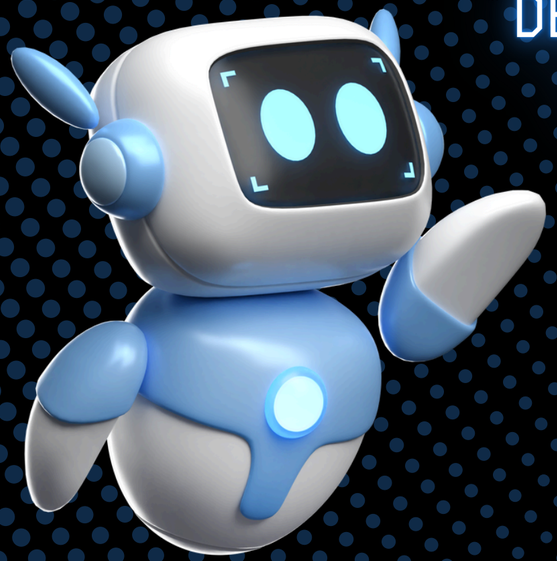
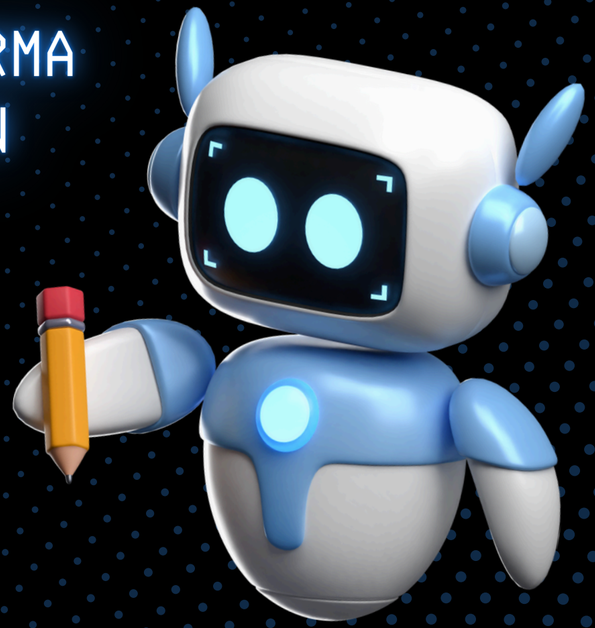


DESPLIEGUE EN CONTENEDORES DE UNA PLATAFORMA BASADA EN MICROSERVICIOS PARA LA GESTIÓN ACADÉMICA.

ESTEFANIA HERNANDEZ ROJAS
NATALIA MORENO MONTOYA
VALENTINA BUENO COLLAZOS
VALERIA FRANCO CORDOBA



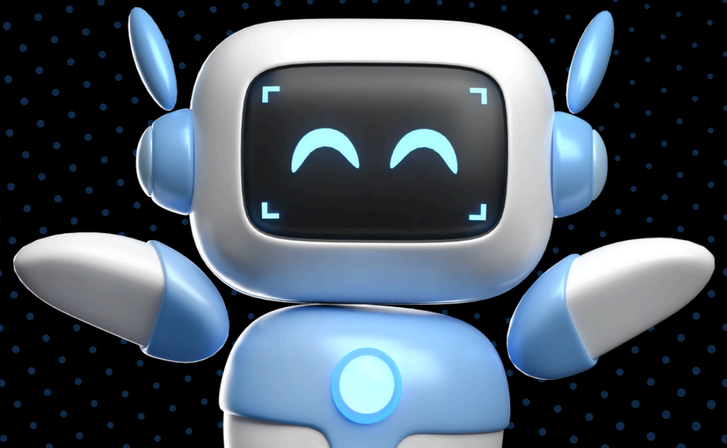
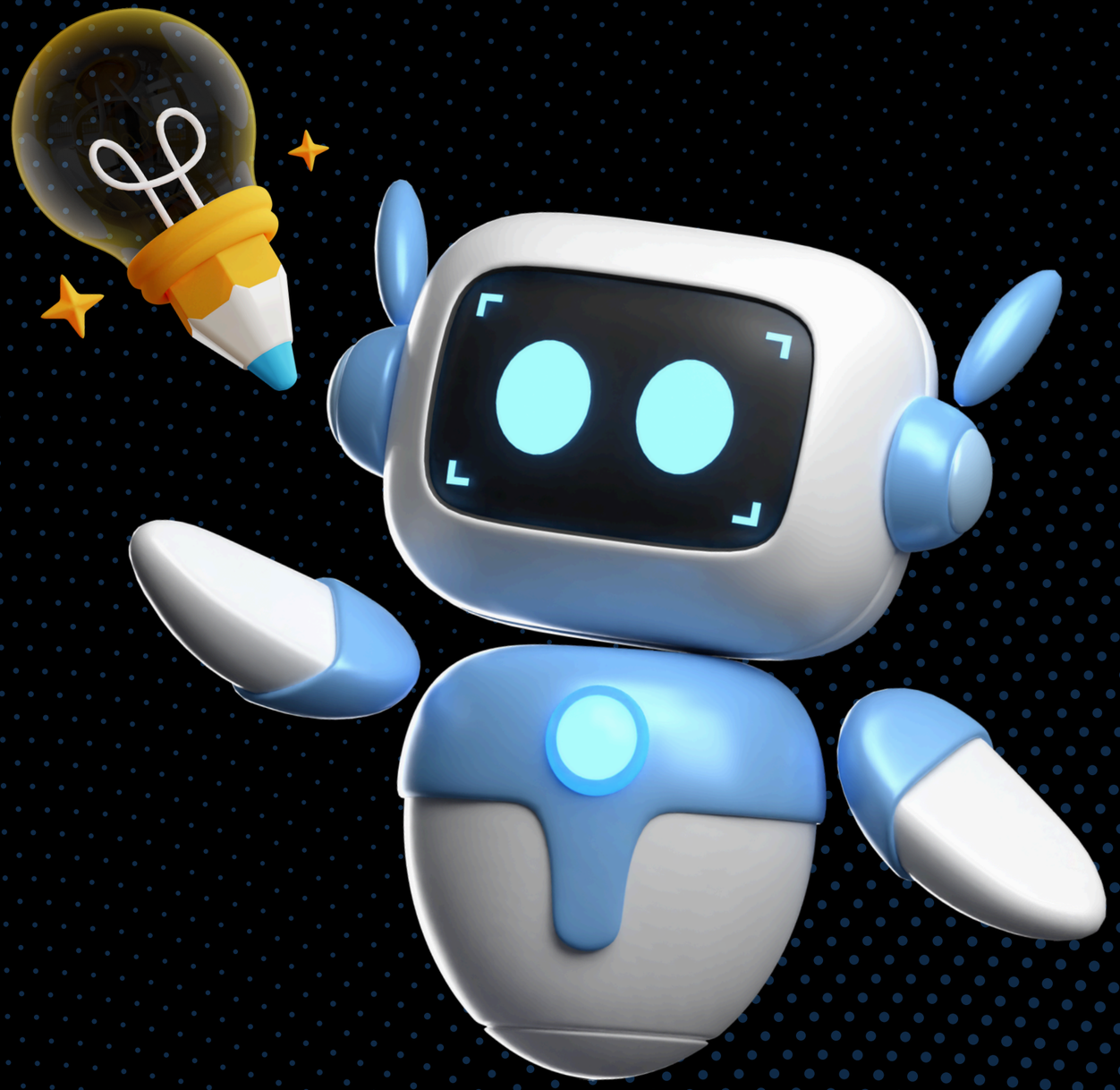
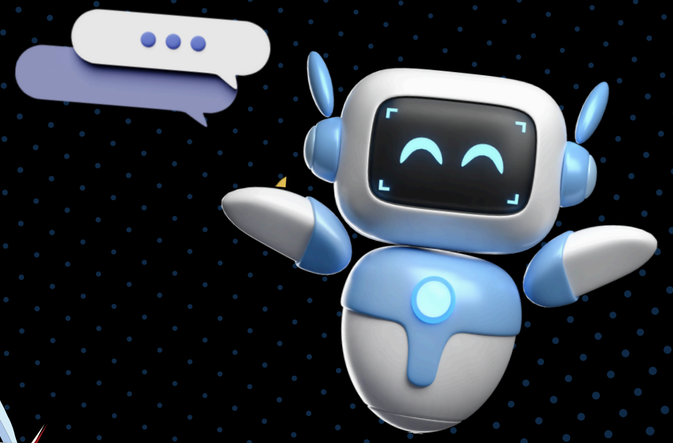


TABLE OF CONTENTS

- Introducción 03
- Análisis 04
- Diseño 08
- Implementación 12
- Pruebas 21
- Conclusión 26



INTRODUCCIÓN



Las instituciones de educación superior enfrentan retos cada vez más complejos en la gestión eficiente de su información.

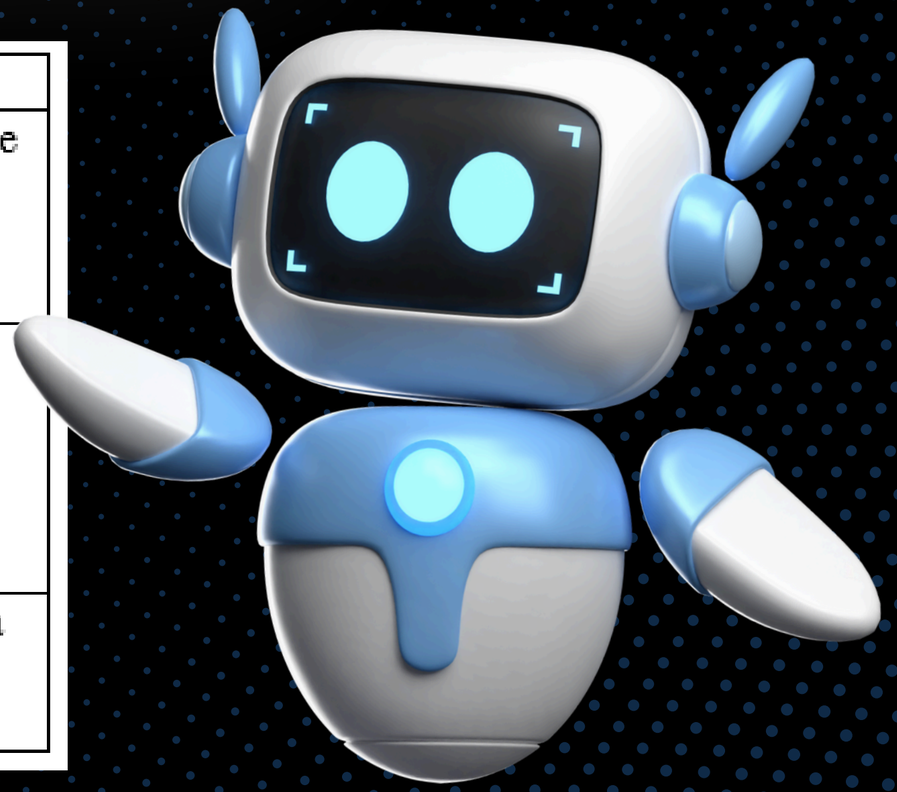
La sobrecarga puede afectar la calidad del servicio educativo y la eficiencia de la administración académica de los programas.

Aplicación de microservicios para gestionar un programa académico, permitiendo administrar estudiantes, profesores y cursos, con despliegue en un clúster distribuido para escalabilidad y rendimiento óptimo.

ANÁLISIS

ALTERNATIVAS DE SOLUCIÓN

Alternativas	Descripción
Docker Compose + Docker Swarm + HA Proxy (Least Connection) JMeter	Cumple su propósito para proyectos de mediano a grande, tiene las características necesarias para el proyecto.
Docker Compose + Kubernetes + Ngix (Round robin) + JMeter	Kubernetes ofrece mejor manejo para aplicaciones distribuidas, pero es más complejo de gestionar y configurar, además, requiere más recursos de cómputo.
Docker Compose + Apache Mesos + HA Proxy (Round robin) + Artillery	Apache Mesos es muy específico para ambientes de Bigdata, por lo que no iba enfocado al proyecto.



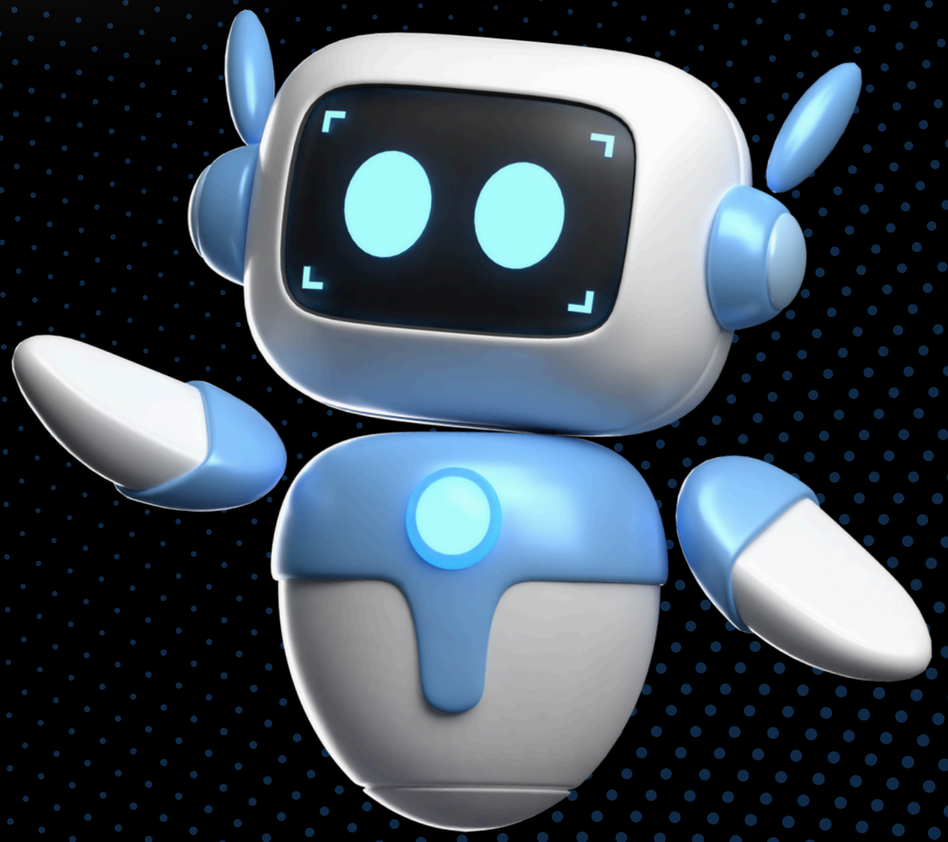
ALTERNATIVAS - CLÚSTER

Clúster de procesamiento:

- Apache Hadoop
- Apache Spark

Visualización de los datos:

- PowerBI
- Python
- GraphX



DEFINICIÓN DE LA ARQUITECTURA

01

Microservicios y contenedores (docker compose y docker swarm)

02

API REST

03

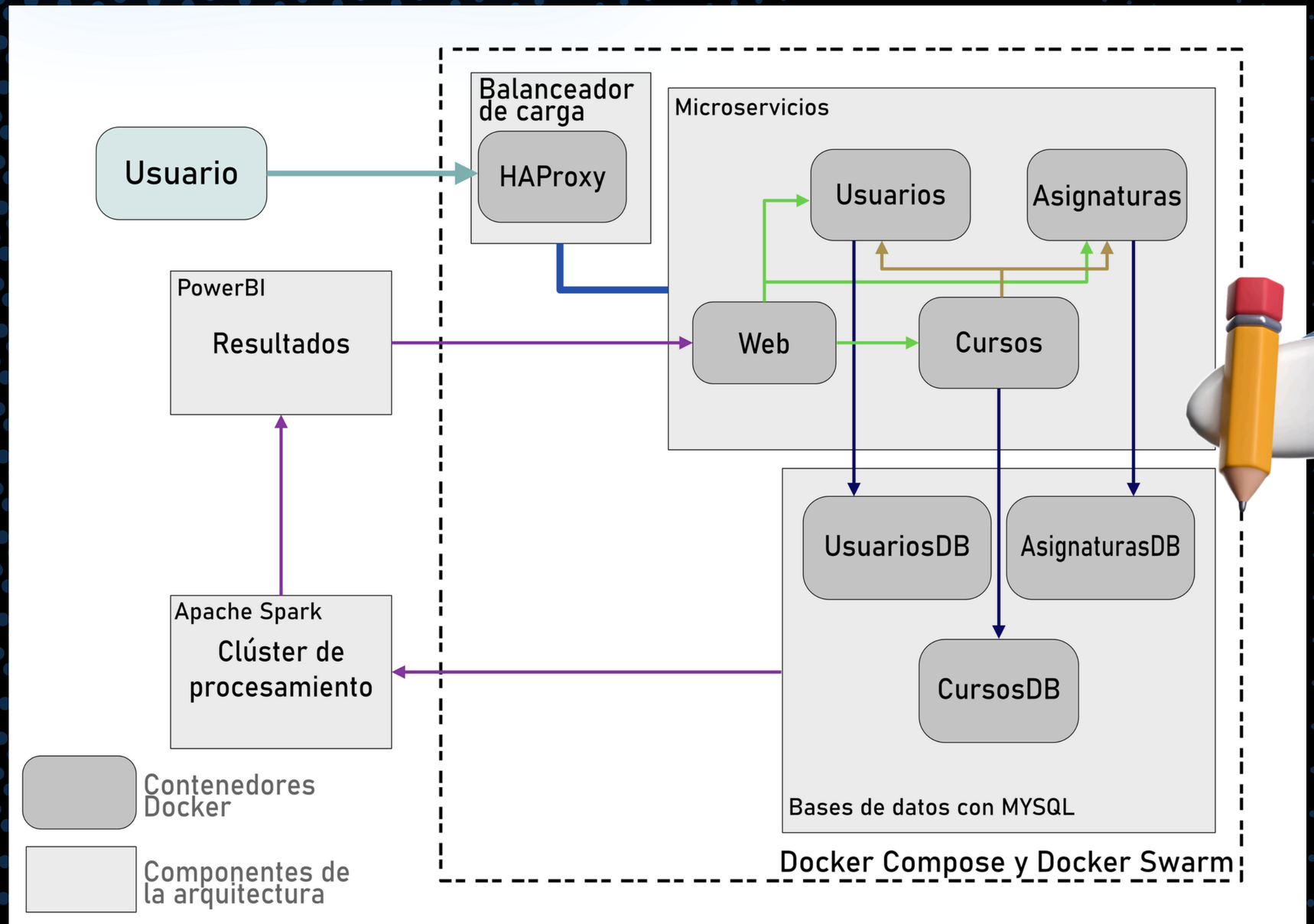
Orquestación (Docker Swarm) y balanceo de carga (HAProxy/least Connection)

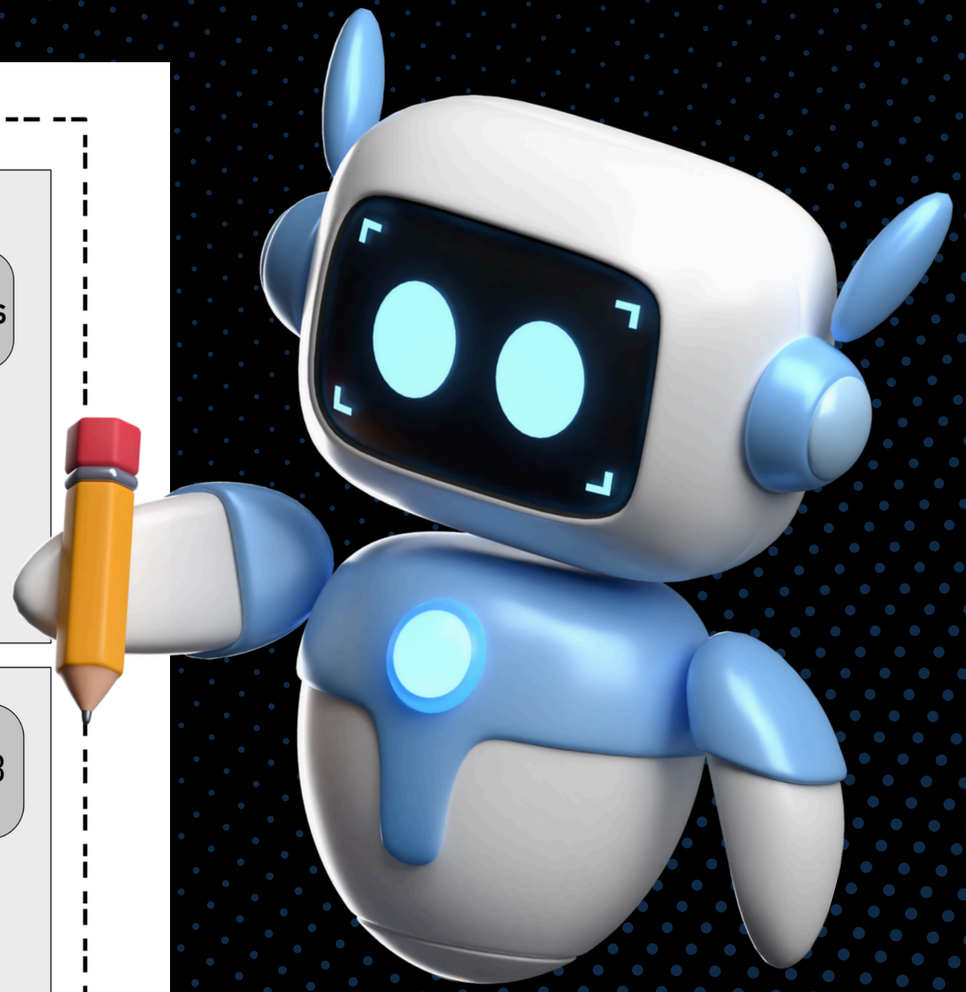
04

Clúster de procesamiento de datos (Apache Spark y PowerBI)

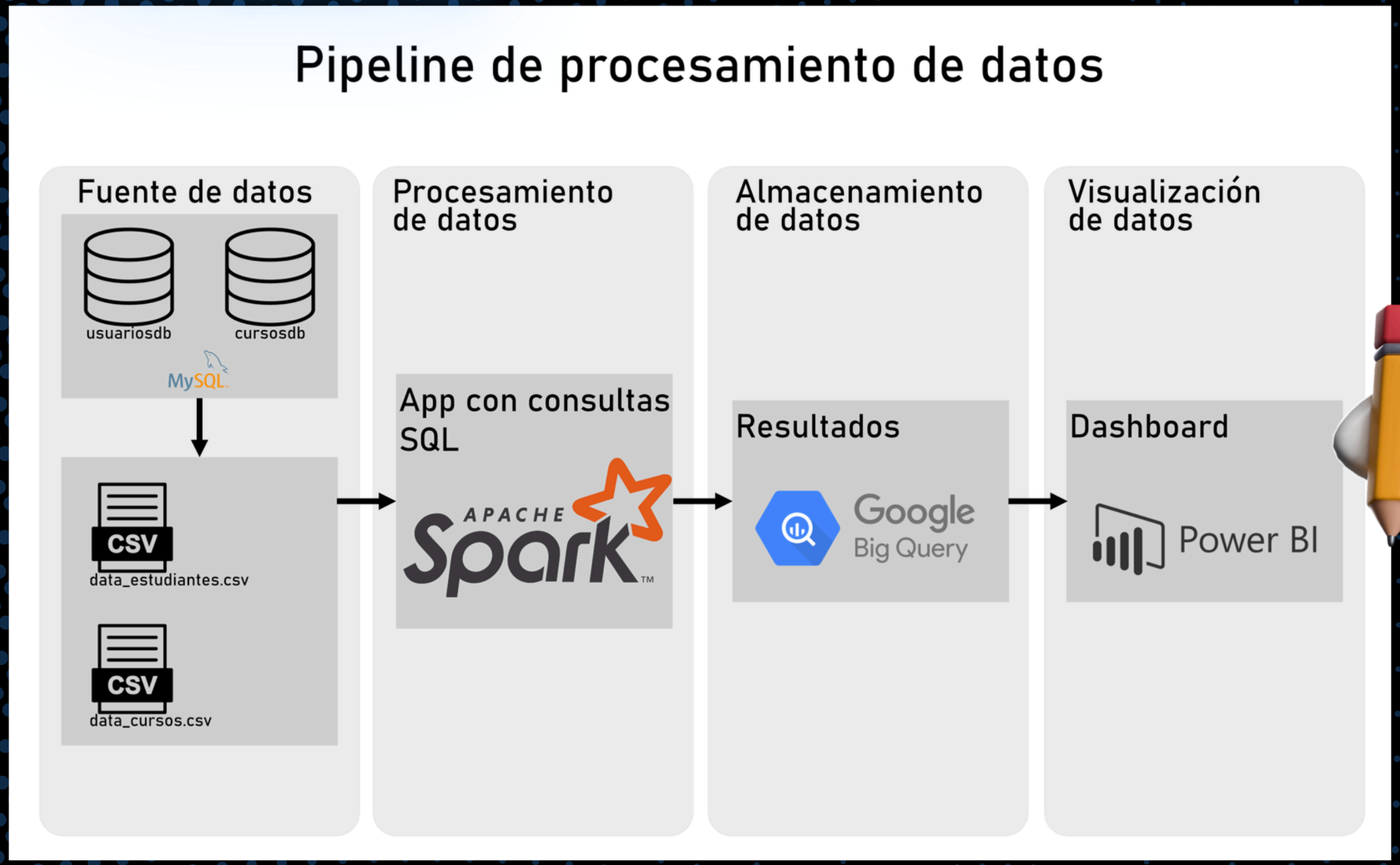
DISEÑO

ARQUITECTURA





PIPELINE


Pipeline de procesamiento de datos
Fuente de datos
usuariosdb
cursosdb
MySQL
CSV
data_estudiantes.csv
CSV
data_cursos.csv
Procesamiento de datos
App con consultas SQL
APACHE Spark
Almacenamiento de datos
Resultados
Google Big Query
Visualización de datos
Dashboard
Power BI

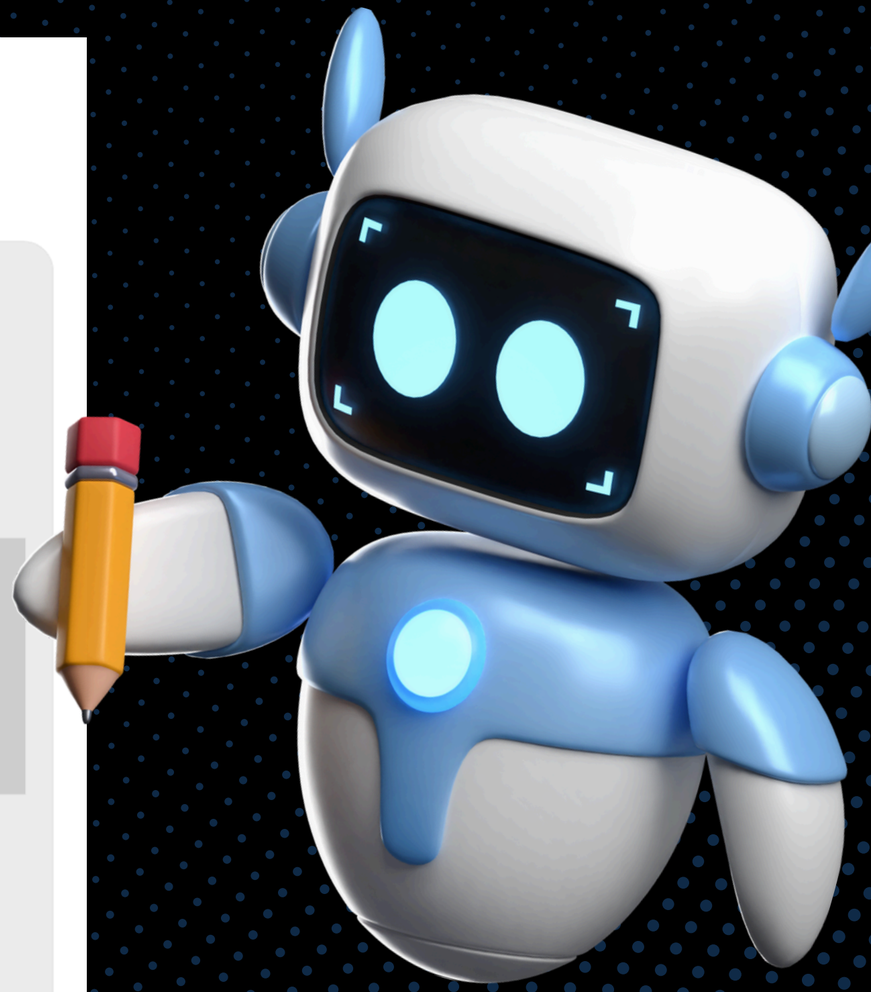
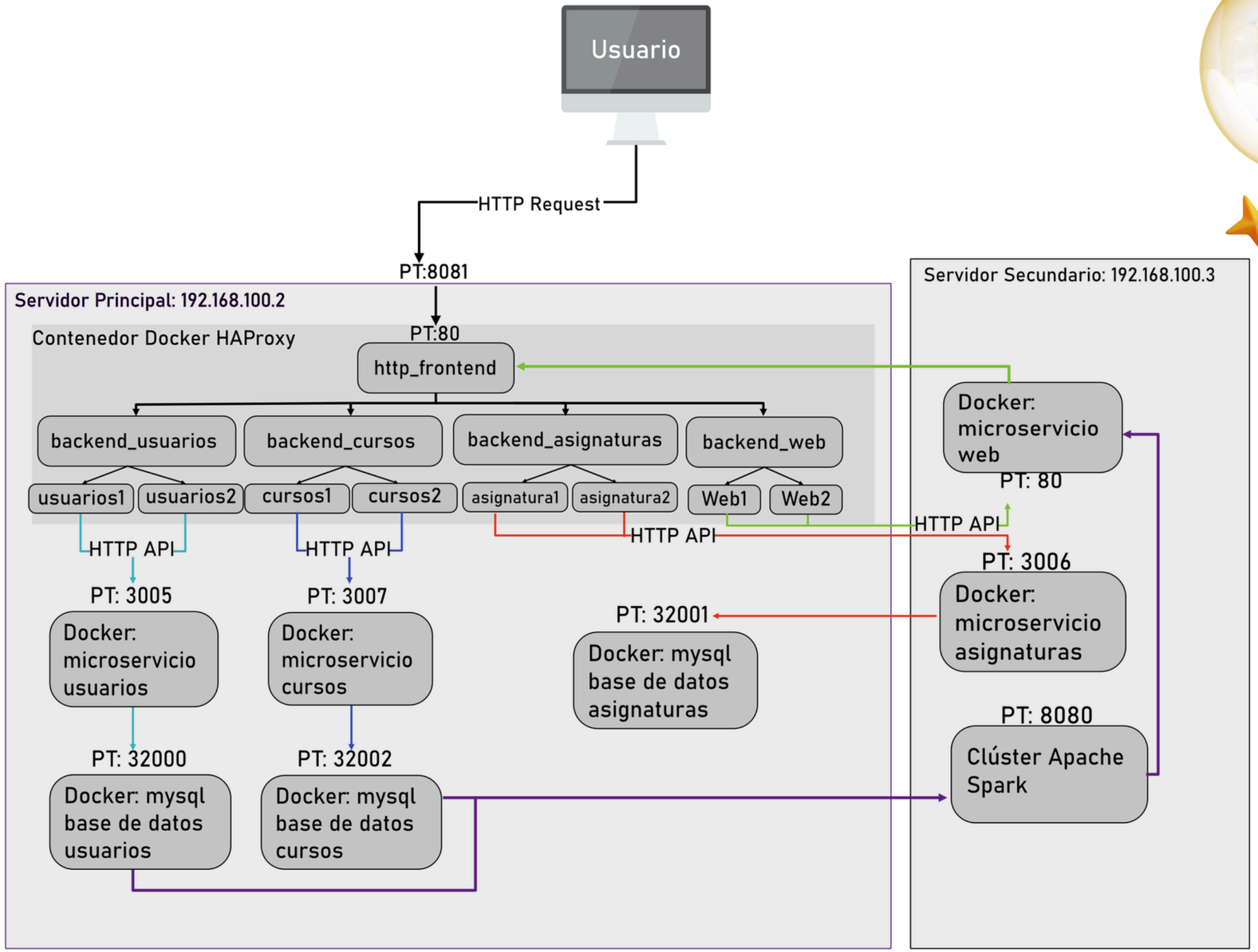
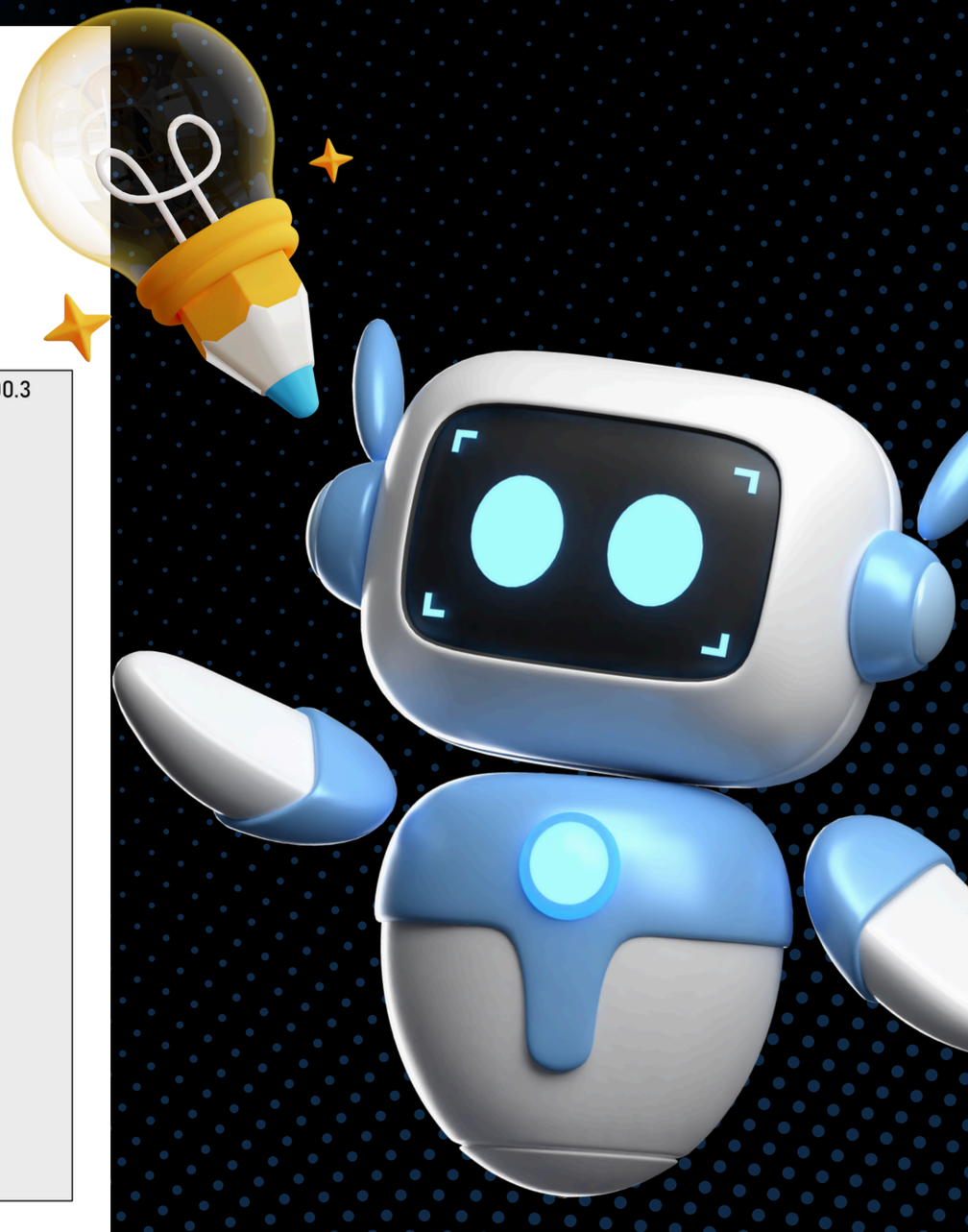


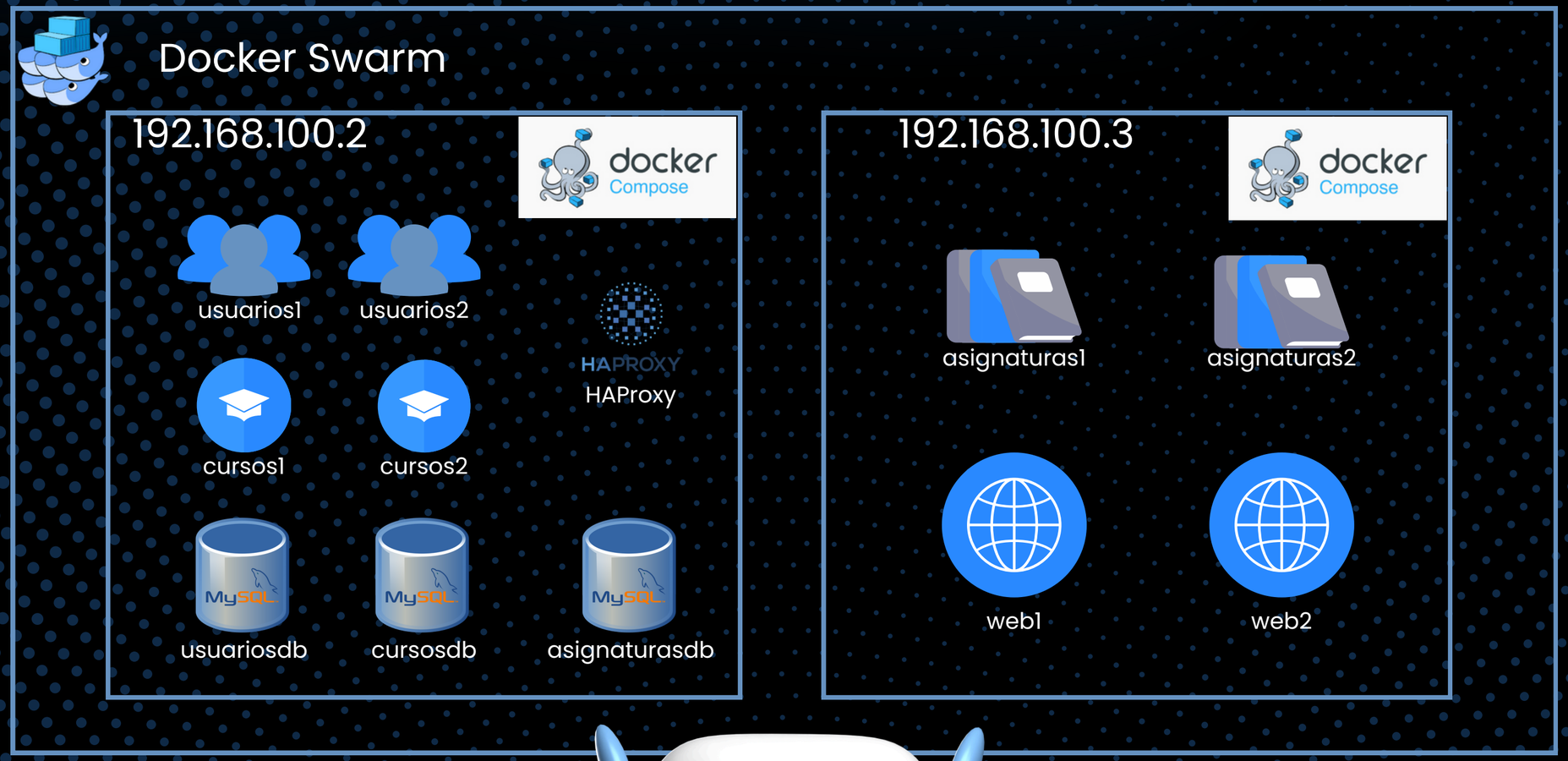
DIAGRAMA DE DESPLIEGUE





IMPLEMENTACIÓN

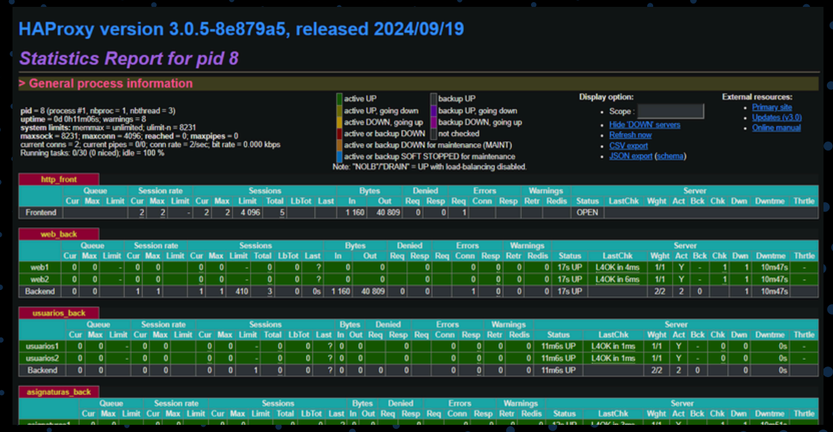
EMPAQUETADO Y DISTRIBUCION



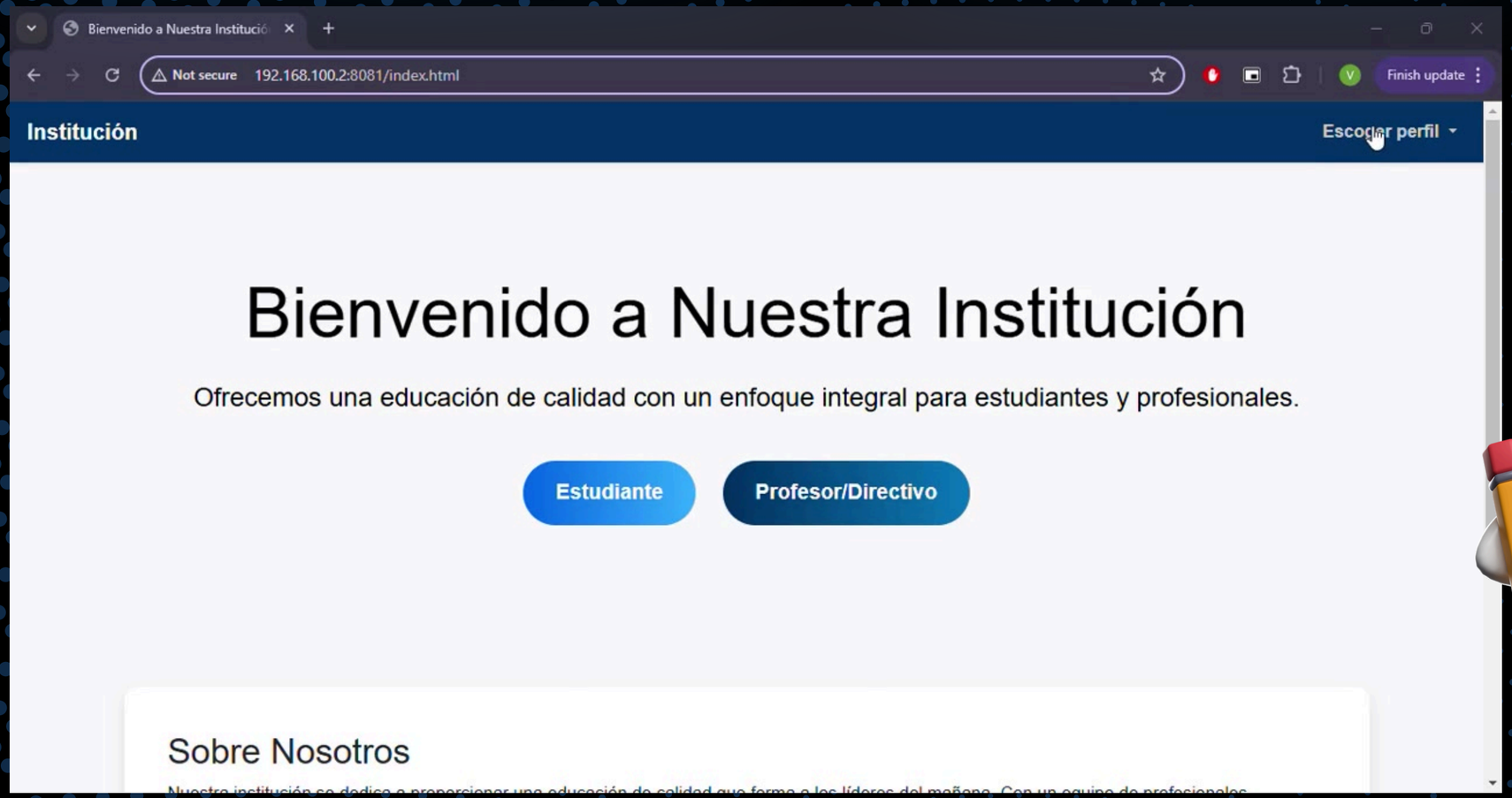
BALANCEO DE CARGA

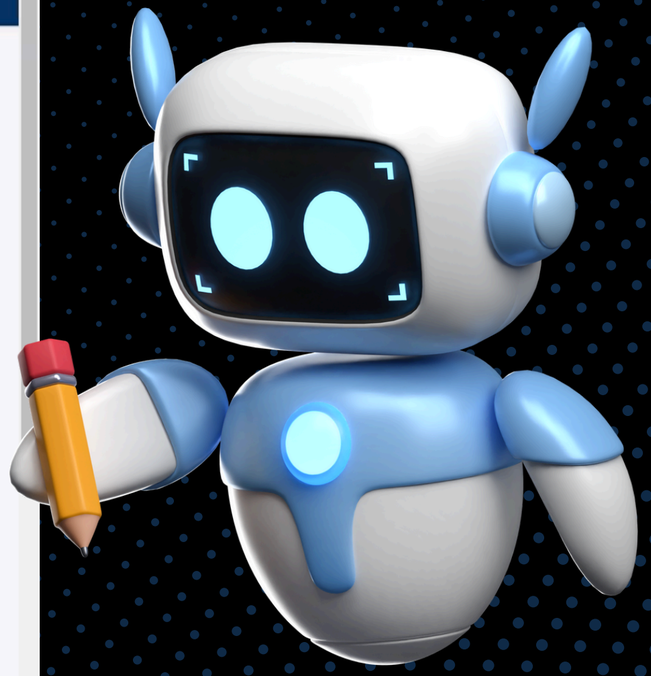
LEASTCONN

BALANCEO DE CARGA

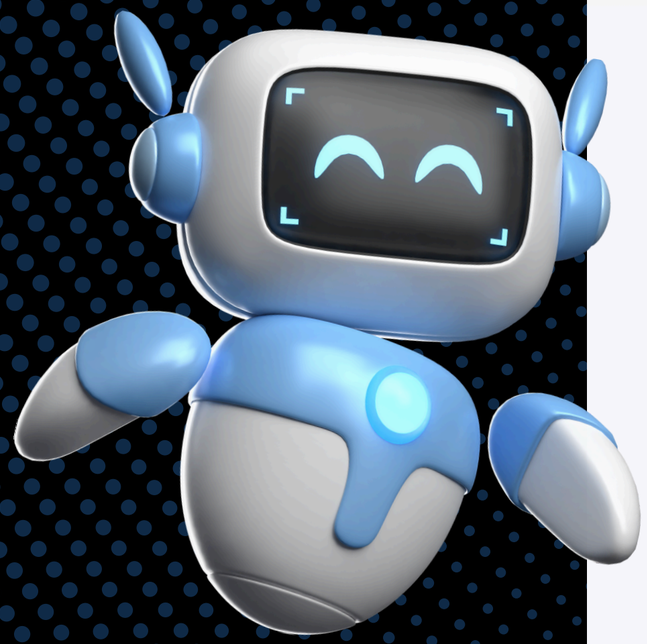


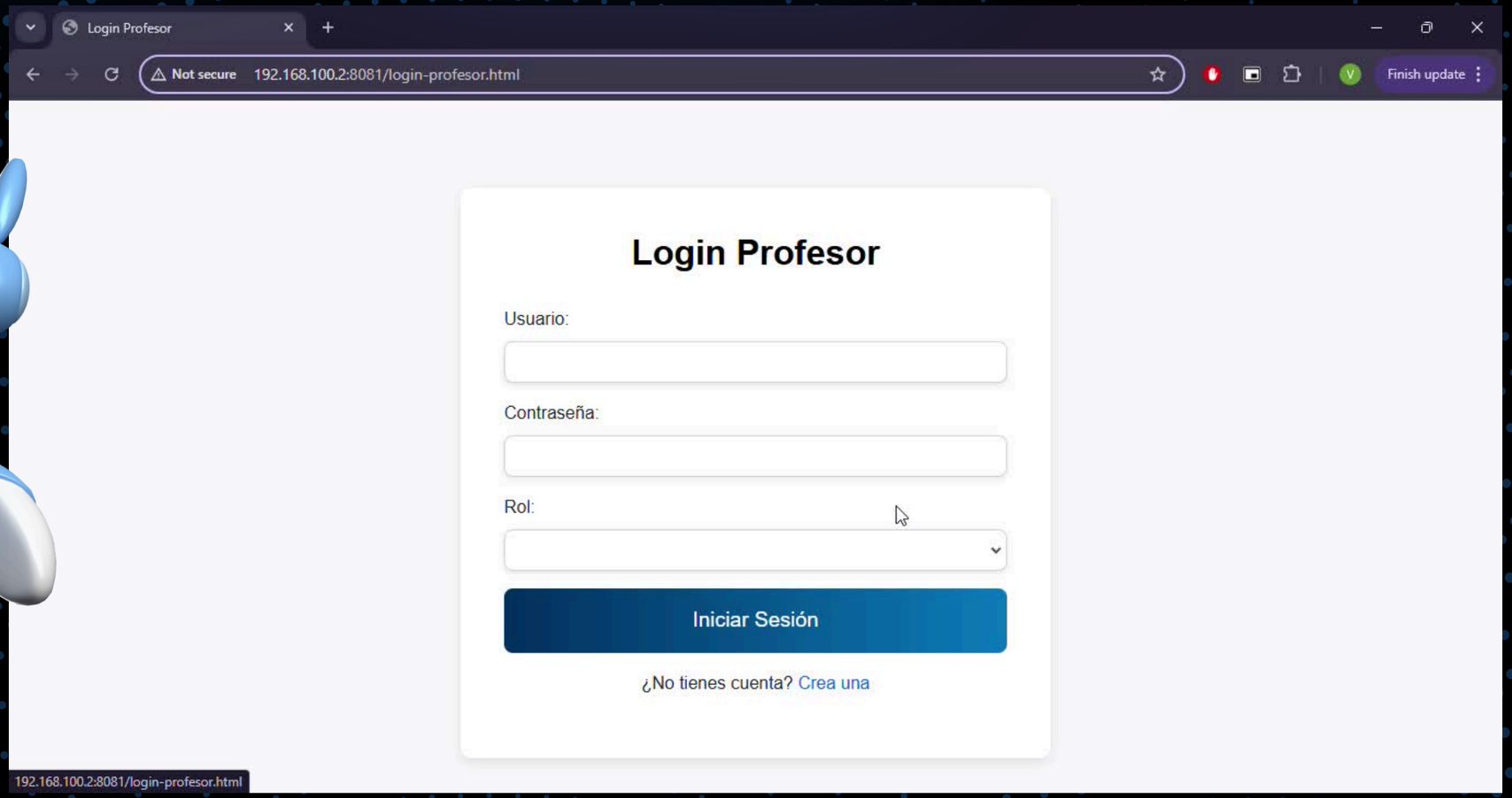
FUNCIONAMIENTO - ESTUDIANTES



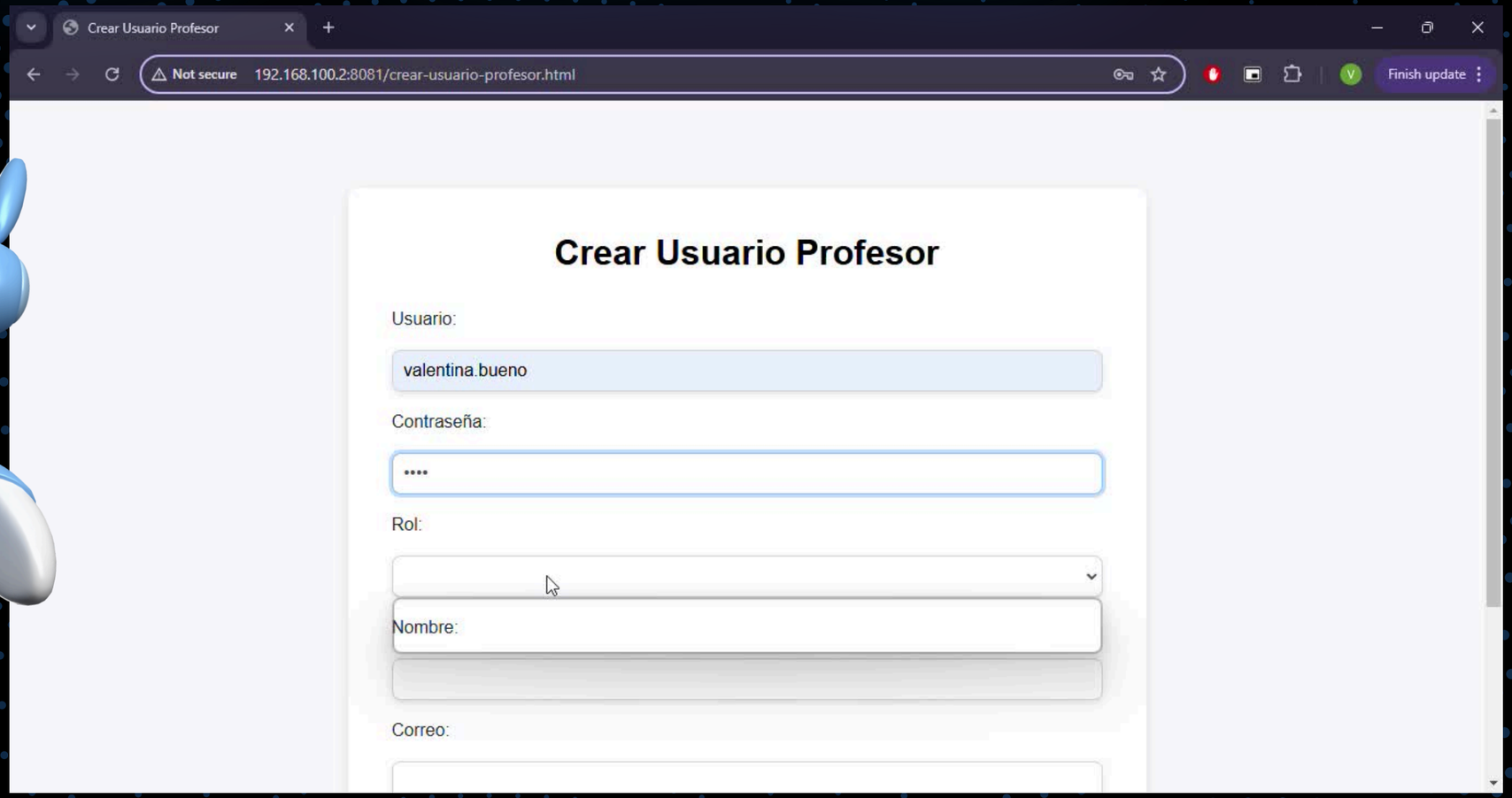


FUNCIONAMIENTO - PROFESORES





FUNCIONAMIENTO - ADMINISTRADOR



CLUSTER DE PROCESAMIENTO DE DATOS

ETAPA 1

Recepción de Datos en CSV

ETAPA 2

Procesamiento en Apache Spark con PySpark

ETAPA 3

Almacenamiento en BigQuery

ETAPA 4

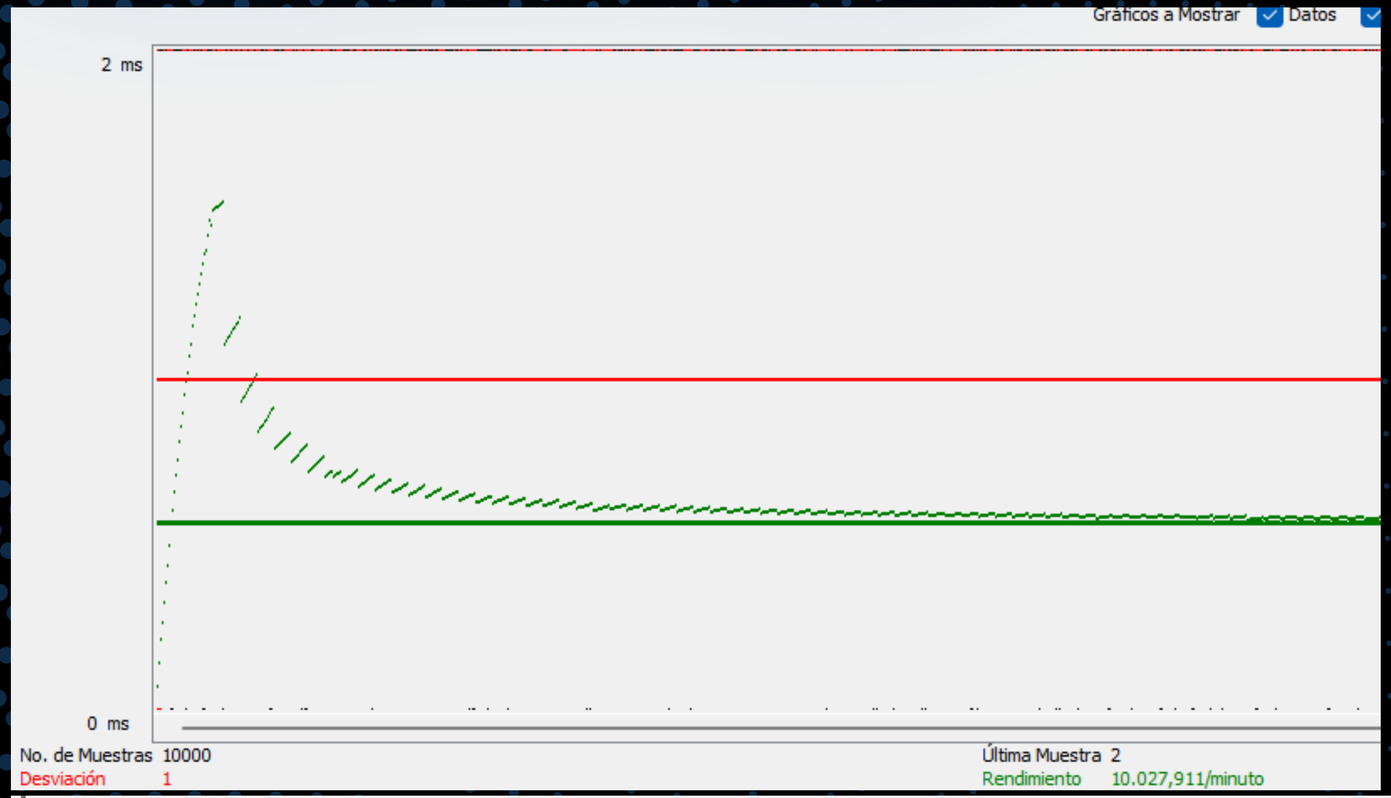
Visualización en PowerBI

FUNCIONAMIENTO DEL CLUSTER

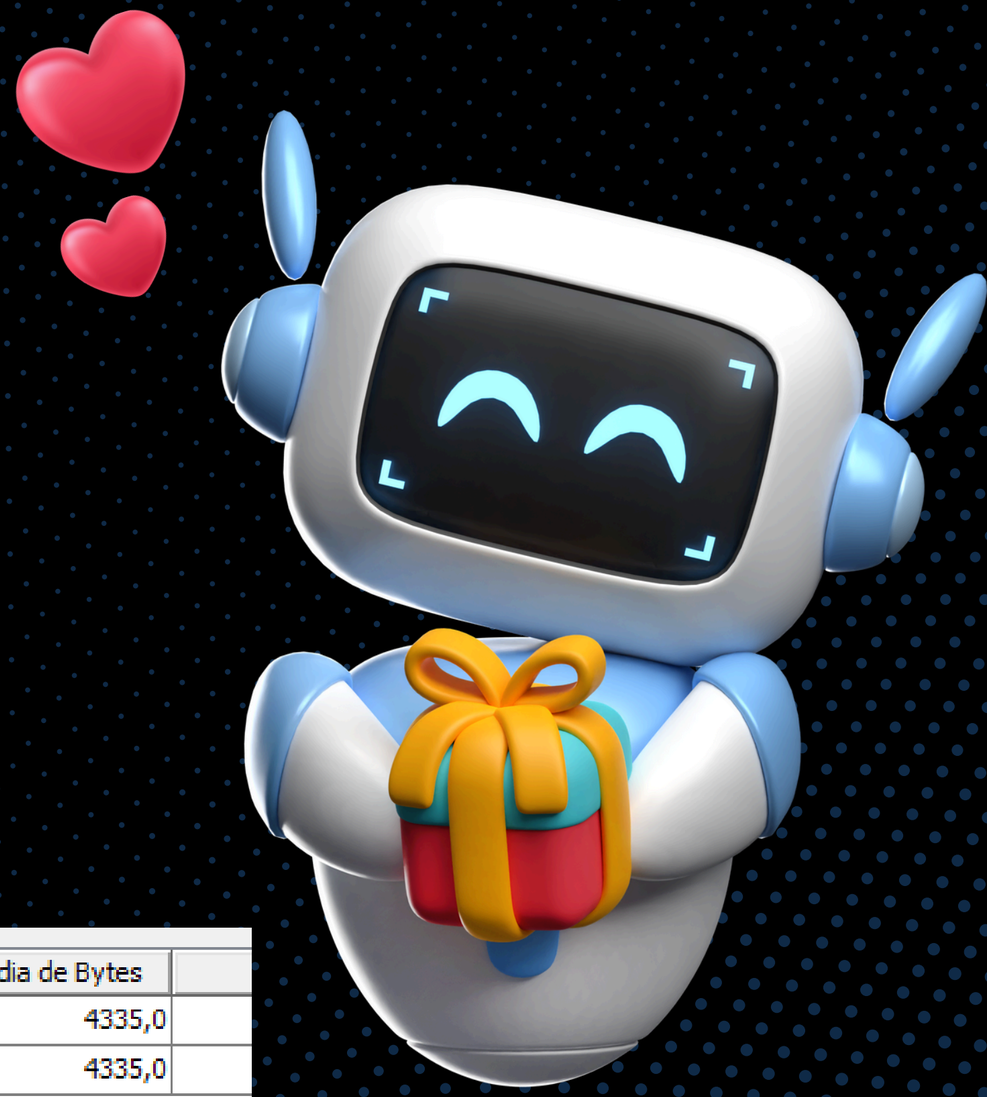
```
root@servidorUbuntu: /home   root@servidorUbuntu: /home/   root@servidorUbuntu: ~/proy

root@servidorUbuntu:/home/vagrant/labSpark/spark-3.5.3-bin-hadoop3/sbin# ./start-worker.sh spark://192.168.100.2:7077
starting org.apache.spark.deploy.worker.Worker, logging to /home/vagrant/labSpark/spark-3.5.3-bin-hadoop3/logs/spark-root-org.apache.spark.deploy.worker.Wor
ker-1-servidorUbuntu.out
root@servidorUbuntu:/home/vagrant/labSpark/spark-3.5.3-bin-hadoop3/sbin# cd ..
root@servidorUbuntu:/home/vagrant/labSpark/spark-3.5.3-bin-hadoop3# cd bin/
root@servidorUbuntu:/home/vagrant/labSpark/spark-3.5.3-bin-hadoop3/bin# ./spark-submit --master spark://192.168.100.2:7077 --conf spark.executor.memory=1g /
root/proyecto_final_redes/clusterAplicacion/appRendimientoEstudiantes.py
^Croot@servidorUbuntu:/home/vagrant/labSpark/spark-3.5.3-bin-hadoop3/bin#
root@servidorUbuntu:/home/vagrant/labSpark/spark-3.5.3-bin-hadoop3/bin# ./spark-submit --master spark://192.168.100.2:7077 --conf spark.executor.memory=1g /
root/proyecto_final_redes/clusterAplicacion/appRendimientoEstudiantes.py
24/11/08 04:23:08 INFO SparkContext: Running Spark version 3.5.3
24/11/08 04:23:08 INFO SparkContext: OS info Linux, 5.15.0-116-generic, amd64
24/11/08 04:23:08 INFO SparkContext: Java version 18.0.2-ea
24/11/08 04:23:09 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/11/08 04:23:10 INFO ResourceUtils: ==============================================================
24/11/08 04:23:10 INFO ResourceUtils: No custom resources configured for spark.driver.
24/11/08 04:23:10 INFO ResourceUtils: ==============================================================
24/11/08 04:23:10 INFO SparkContext: Submitted application: Dashboard Información Estudiantes
24/11/08 04:23:10 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(memory -> name: memory, amount: 1024, script: , vendor: , o
ffHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/11/08 04:23:10 INFO ResourceProfile: Limiting resource is cpu
24/11/08 04:23:10 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/11/08 04:23:11 INFO SecurityManager: Changing view acls to: root
24/11/08 04:23:11 INFO SecurityManager: Changing modify acls to: root
24/11/08 04:23:11 INFO SecurityManager: Changing view acls groups to:
24/11/08 04:23:11 INFO SecurityManager: Changing modify acls groups to:
24/11/08 04:23:11 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: root; groups with view perm
issions: EMPTY; users with modify permissions: root; groups with modify permissions: EMPTY
24/11/08 04:23:13 INFO Utils: Successfully started service 'sparkDriver' on port 33439.
24/11/08 04:23:13 INFO SparkEnv: Registering MapOutputTracker
24/11/08 04:23:14 INFO SparkEnv: Registering BlockManagerMaster
24/11/08 04:23:14 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/11/08 04:23:14 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/11/08 04:23:14 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/11/08 04:23:14 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-606dcb06-9108-49df-9f9e-ede0a8874d46
24/11/08 04:23:14 INFO MemoryStore: MemoryStore started with capacity 434.4 MiB
24/11/08 04:23:14 INFO SparkEnv: Registering OutputCommitCoordinator
24/11/08 04:23:16 INFO JettyUtils: Start Jetty 192.168.100.2:4040 for SparkUI
24/11/08 04:23:16 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/11/08 04:23:17 INFO StandaloneAppClient$ClientEndpoint: Connecting to master spark://192.168.100.2:7077...
24/11/08 04:23:18 INFO TransportClientFactory: Successfully created connection to /192.168.100.2:7077 after 238 ms (0 ms spent in bootstraps)
```

PRUEBAS

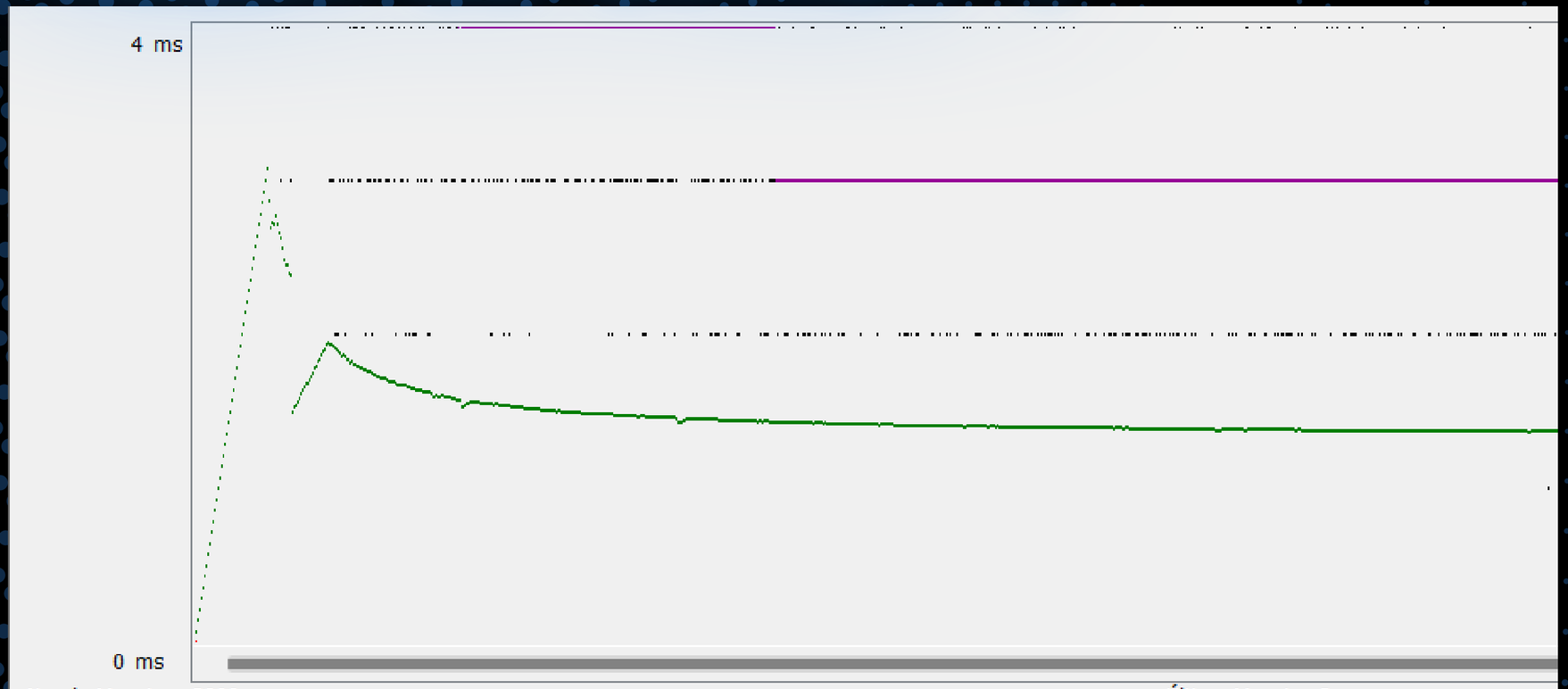
PRUEBA DE CARGA



Etiqueta	# Muestras	Media	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Petición HTTP	10000	1	47	1,50	0,00%	167,1/sec	707,54	29,05	4335,0
Total	10000	1	47	1,50	0,00%	167,1/sec	707,54	29,05	4335,0



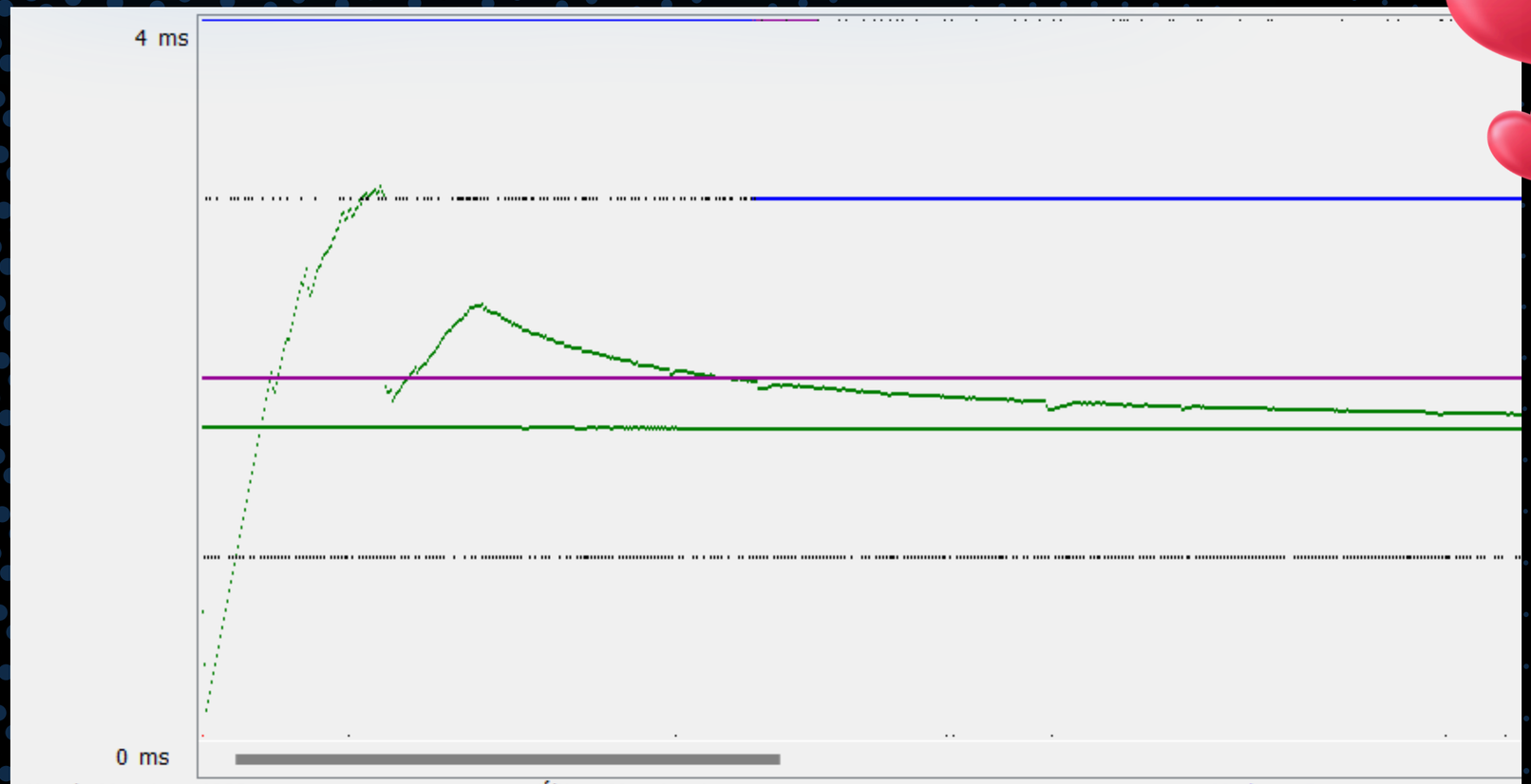
PRUEBA DE ESTRÉS 1



No. de Muestras 2000
Desviación 10
Última Muestra 2
Rendimiento 12.156,823/minuto

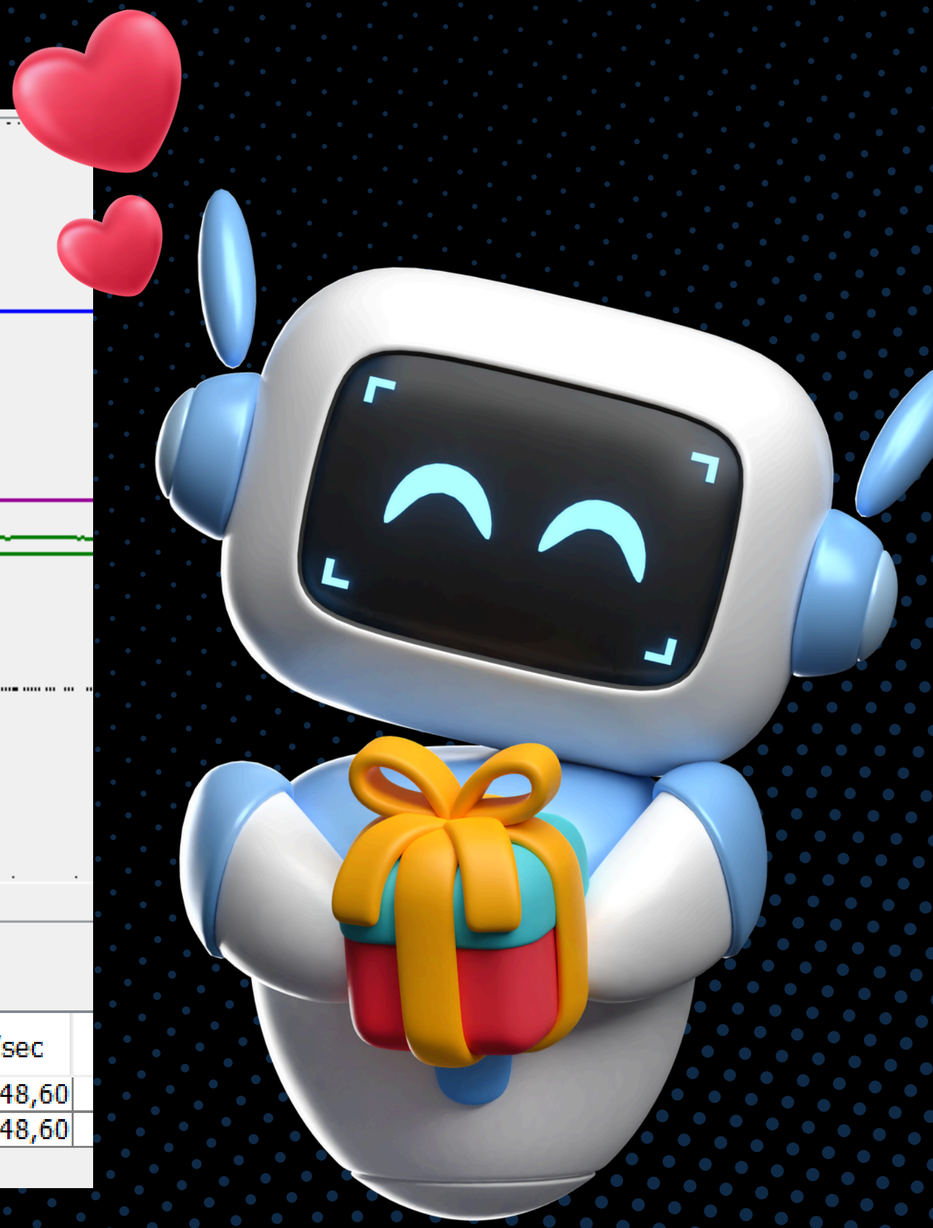
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec
Petición HTTP	2000	4	1	93	10,82	0,00%	202,6/sec	1186,40	24,34
Total	2000	4	1	93	10,82	0,00%	202,6/sec	1186,40	24,34

PRUEBA DE ESTRÉS 2



No. de Muestras 4000 | Última Muestra 1 | Media 3
Desviación 7 | Rendimiento 24.284,124/minuto | Mediana 2

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec
Petición HTTP	4000	3	0	85	7,56	0,03%	404,7/sec	2369,53	48,60
Total	4000	3	0	85	7,56	0,03%	404,7/sec	2369,53	48,60



BALANCEO DE CARGA HAPROXY

http_front

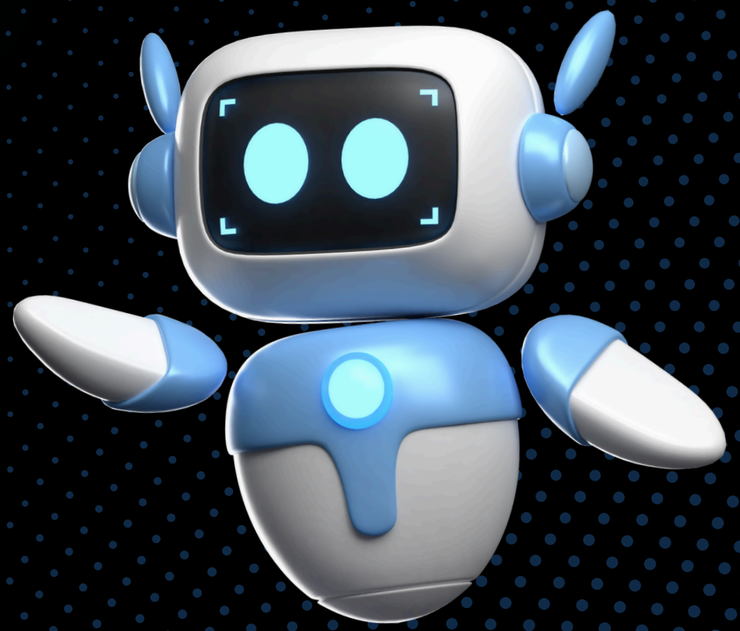
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Stat
Frontend				0	618	-	2	1 847	4 096	43 994			22 507 792	818 207 260	0	0	9					OPE

web_back

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis
web1	0	0	-	0	1 555		0	695	-	88 809	88 809	2m	11 301 243	413 182 068		0		0	117	0	0
web2	0	0	-	0	1 582		0	695	-	88 769	88 769	2m	11 204 818	404 982 351		0		0	102	0	0
Backend	0	0		1	3 137		1	1 390	410	177 582	177 578	0s	22 507 792	818 207 260	0	0		0	219	0	0

CONCLUSIONES

Este proyecto ofrece una solución eficiente para la gestión académica mediante una arquitectura de microservicios, desplegada en contenedores y orquestada, logrando escalabilidad y alto rendimiento. El clúster de procesamiento y análisis de datos, permite gestionar grandes volúmenes de información y generar informes detallados sobre el rendimiento académico, accesibles a través de Power BI.



THANK
YOU!