Sistemas Distribuidos

Semana 10

Jose Emilio Martinez Urgelles

Universidad Cooperativa de Colombia

12/04/2025

Harold Bolaños

**Cali**

Ejercicios de Implementacion

### 1. Desplegar Memcached en Docker

docker run -d \

--name memcached \

-p 11211:11211 \

memcached:alpine

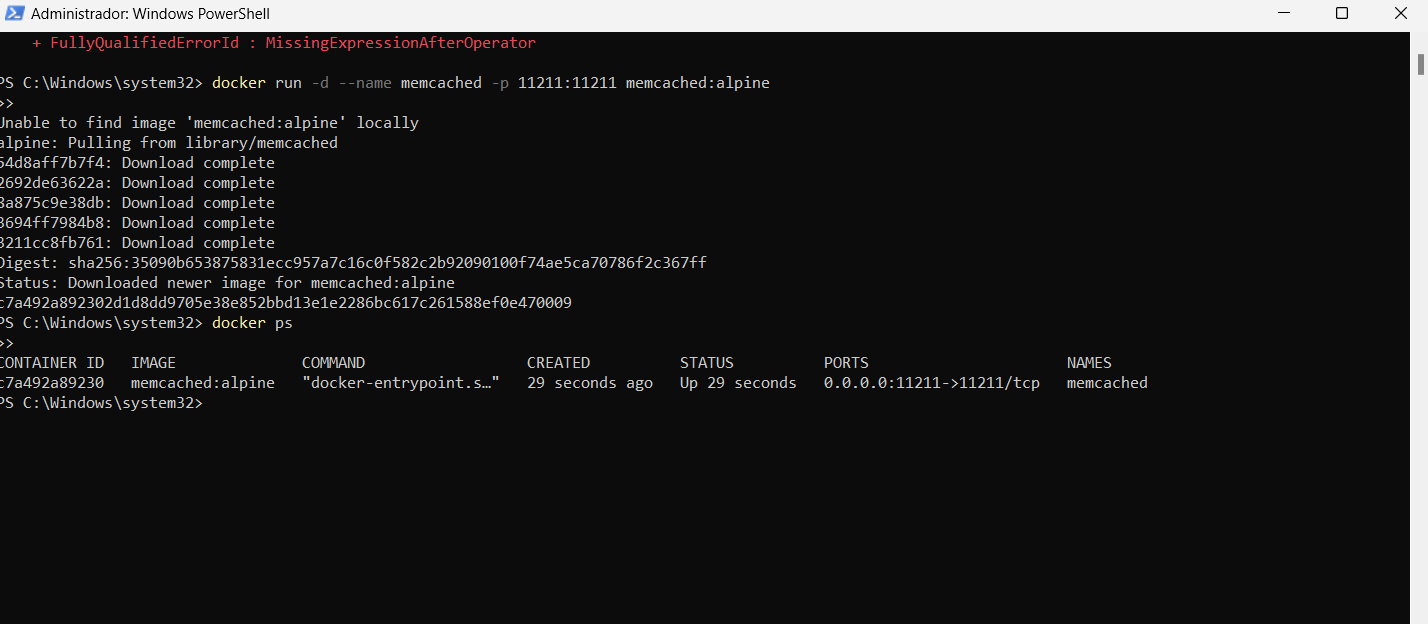
**Verificación (opcional):**

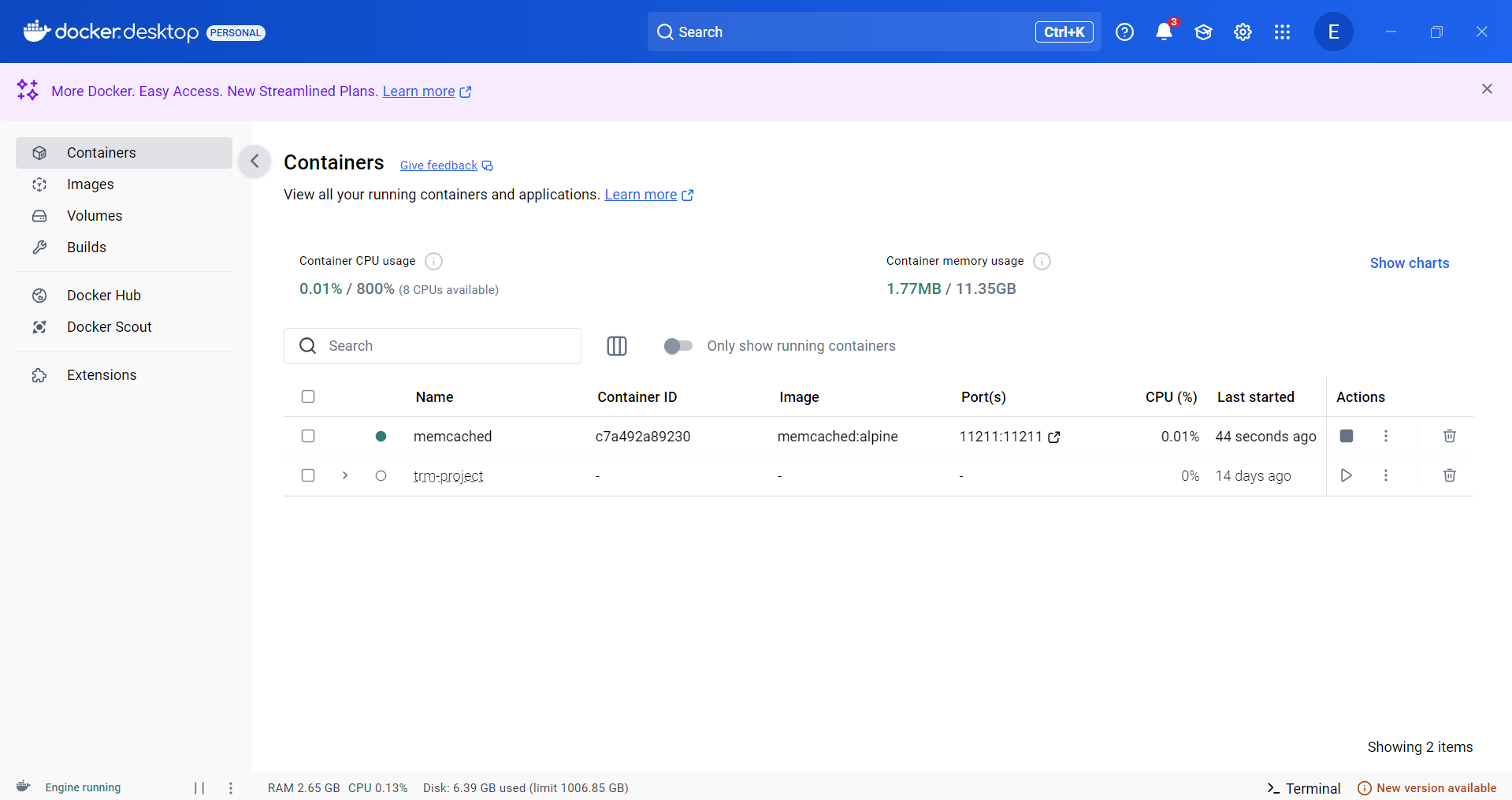
echo -e "set mykey 0 900 4

test" | nc localhost 11211

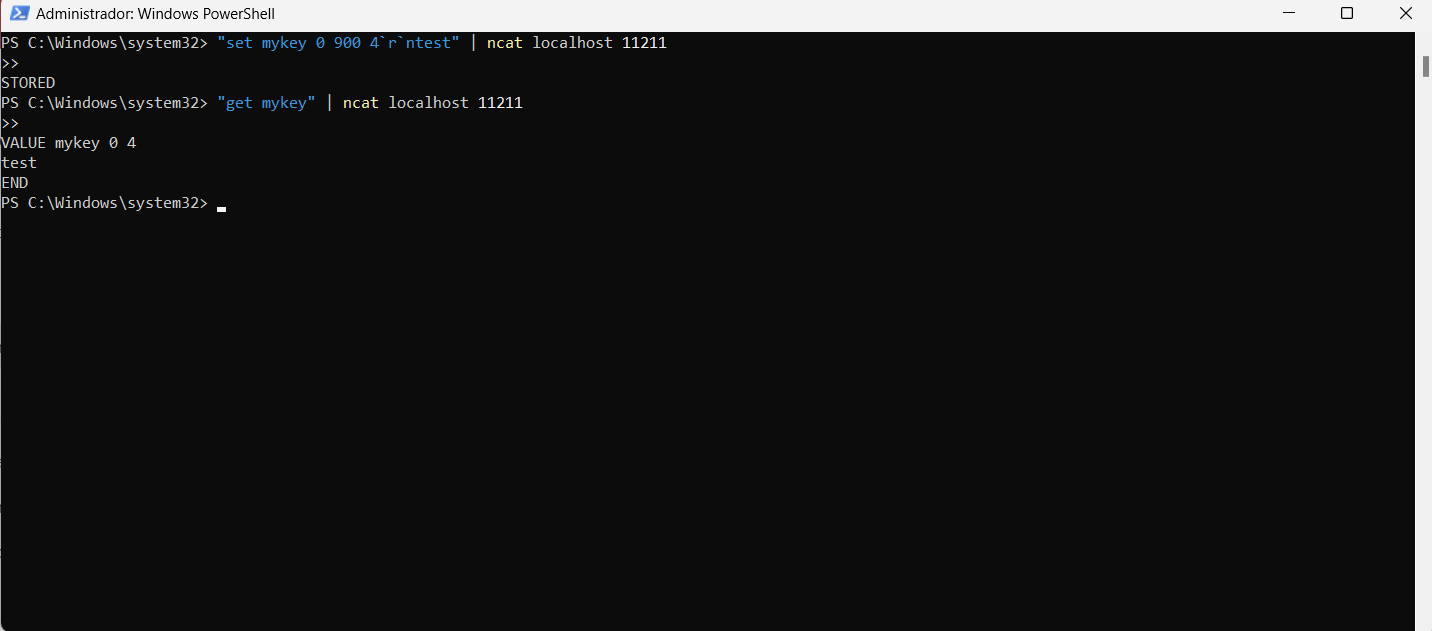
echo -e "get mykey" | nc localhost 11211

Pruebas de Implementacion





Verificacion



### 2. Desplegar Redis en Docker

docker run -d \

--name redis \

-p 6379:6379 \

redis:alpine

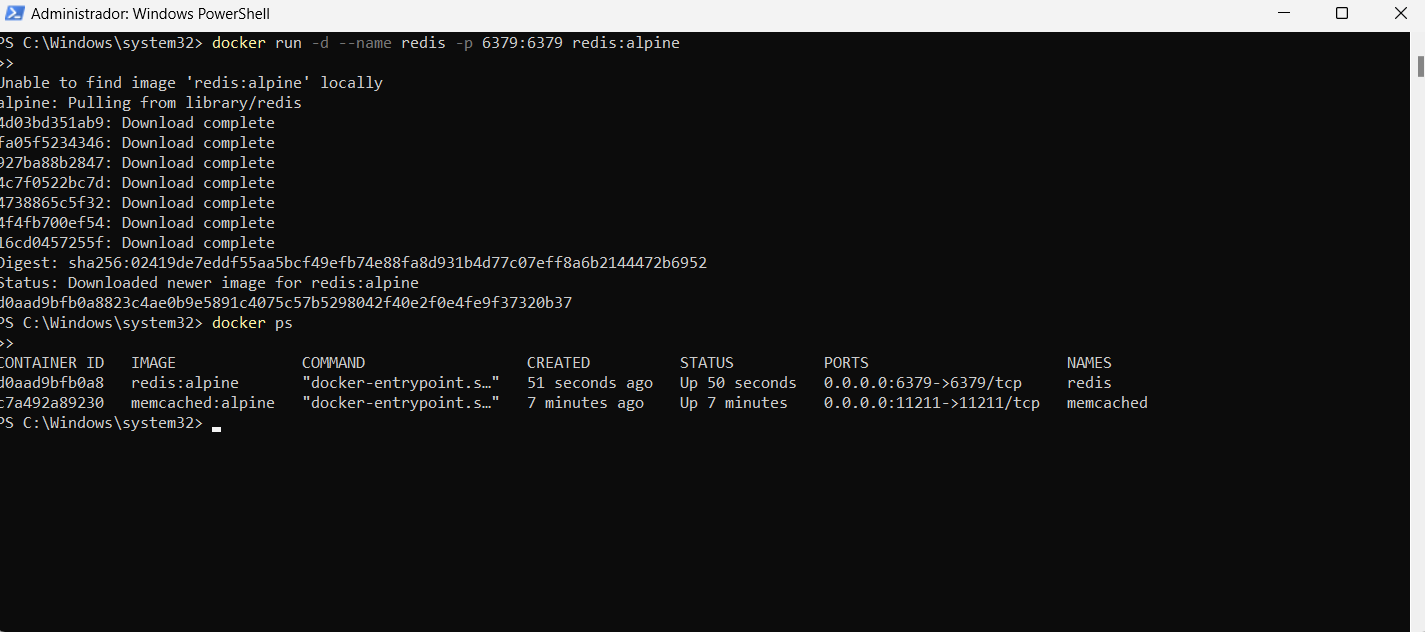
**Verificación (opcional):**

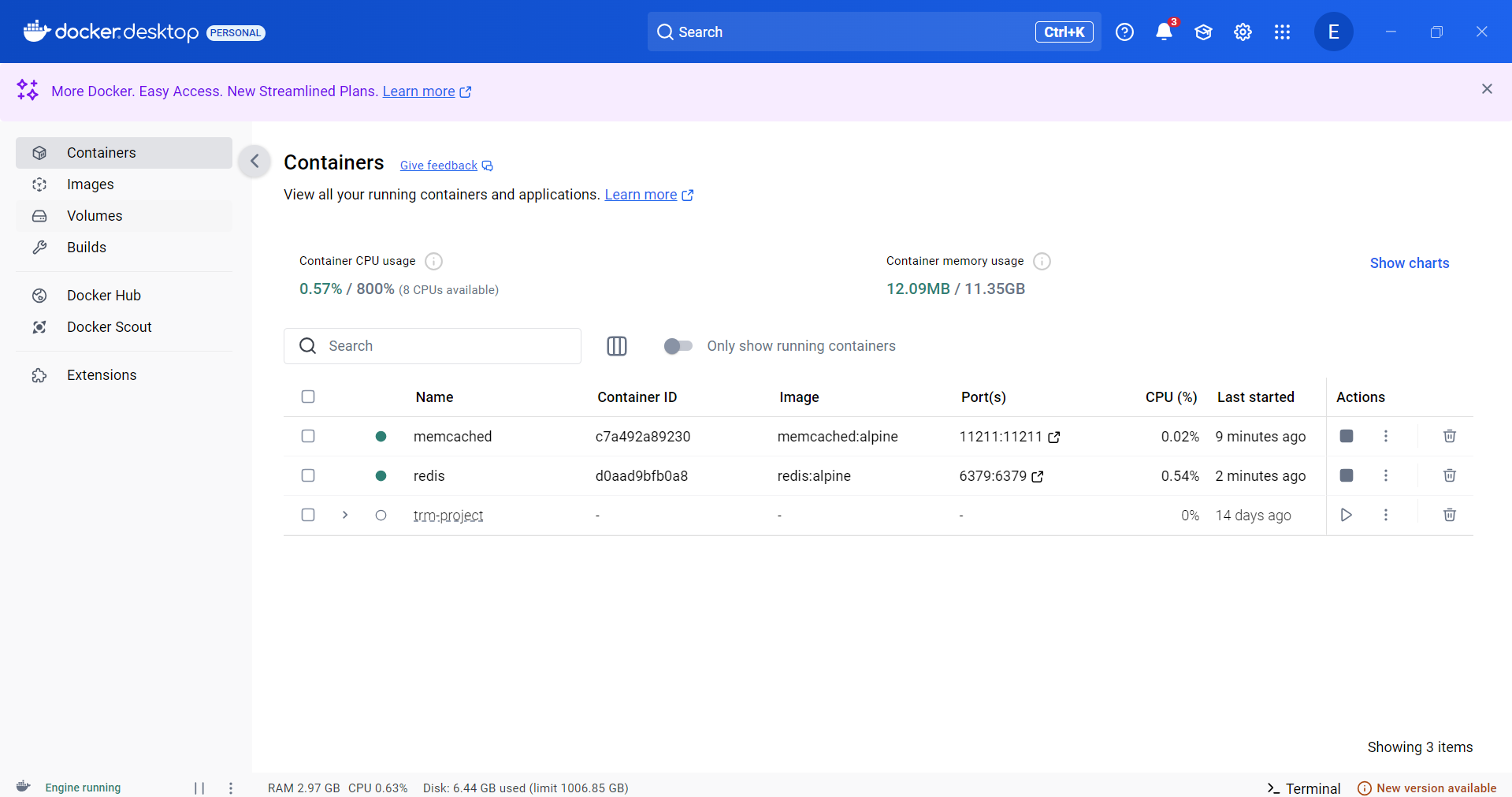
docker exec -it redis redis-cli

> SET testkey "Hello Redis"

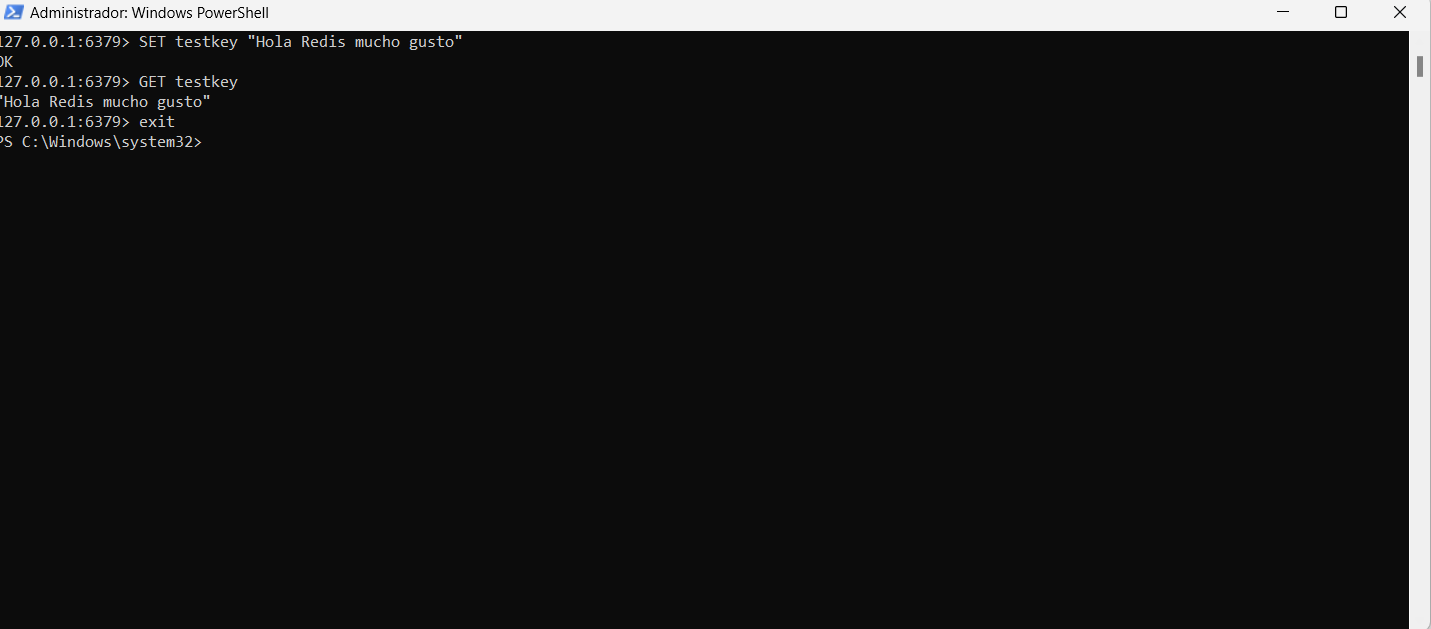
> GET testkey

Prueba de Implementacion





Verificacion



### 3. Desplegar Hazelcast en Docker (modo standalone)

docker run -d \

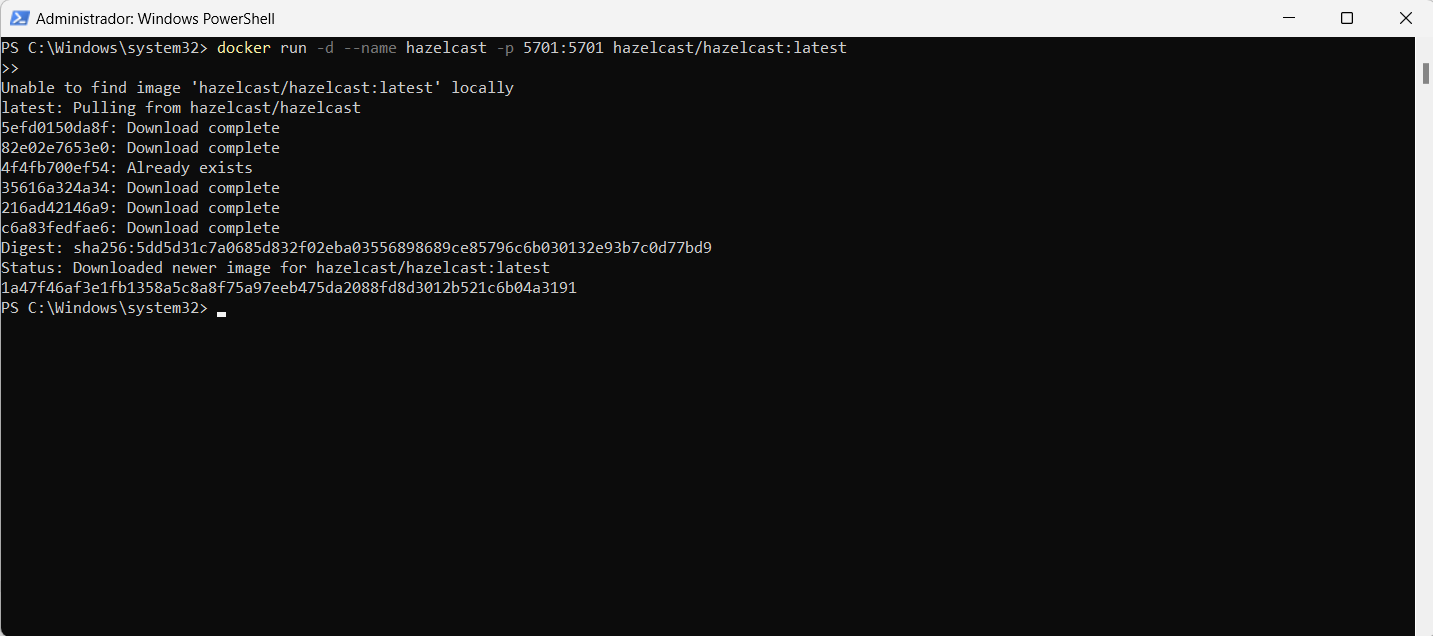
--name hazelcast \

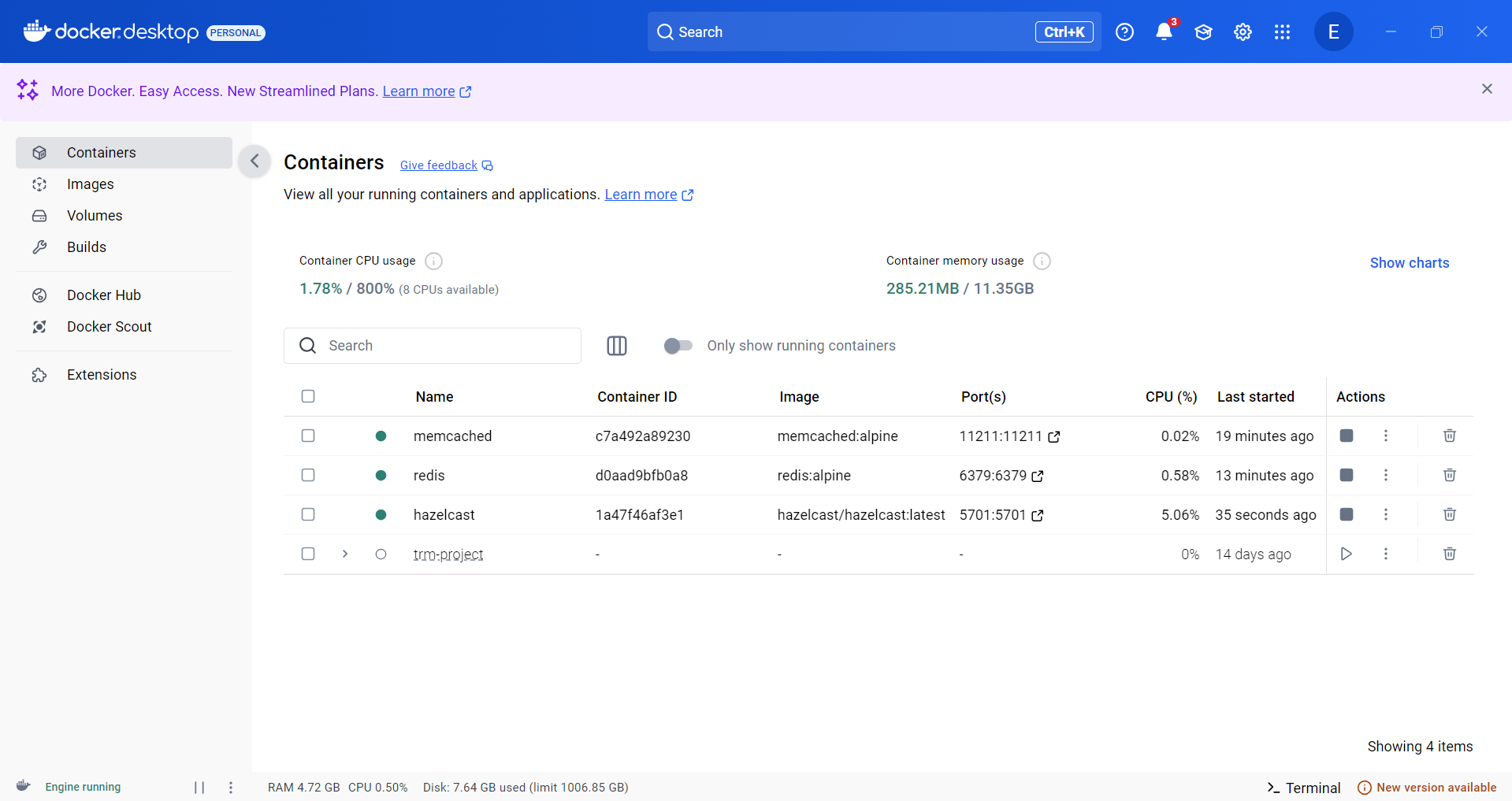
-p 5701:5701 \

hazelcast/hazelcast:latest

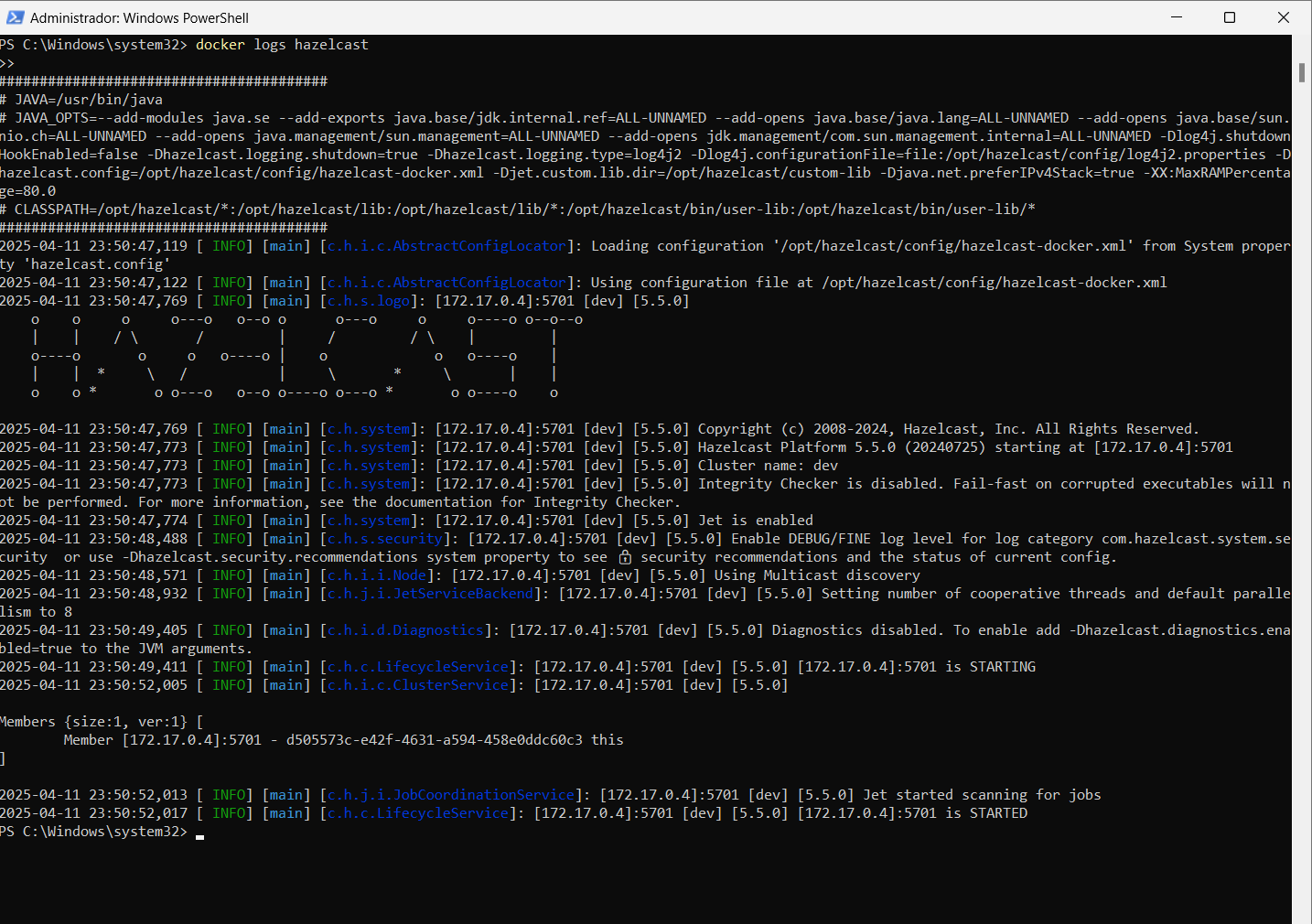
**Verificación (opcional):** Visita http://localhost:5701 (requiere abrir puerto y configurar el dashboard si se desea habilitar la interfaz web).

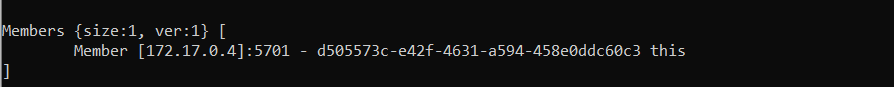
Prueba de Implementacion





Verificacion





Taller Practico modelos de Coherencia

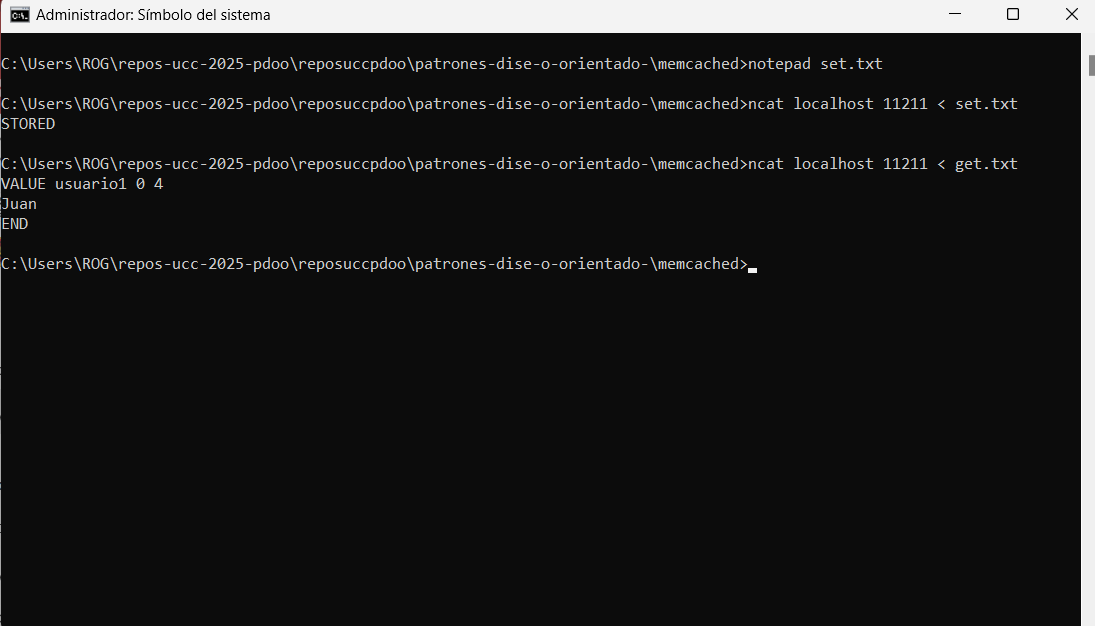
1. **Guardar y recuperar un valor simple**

echo -e "set usuario1 0 900 5

Juan" | nc localhost 11211

echo -e "get usuario1" | nc localhost 11211

Prueba de Implementacion



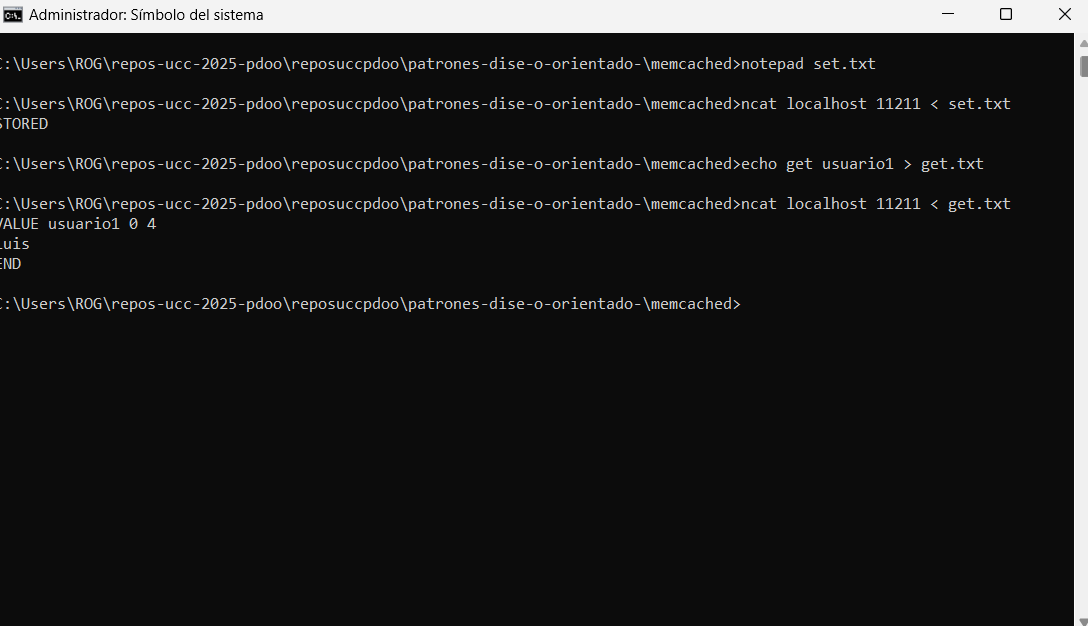
1. **Sobrescribir un valor existente**

echo -e "set usuario1 0 900 4

Luis" | nc localhost 11211

echo -e "get usuario1" | nc localhost 11211

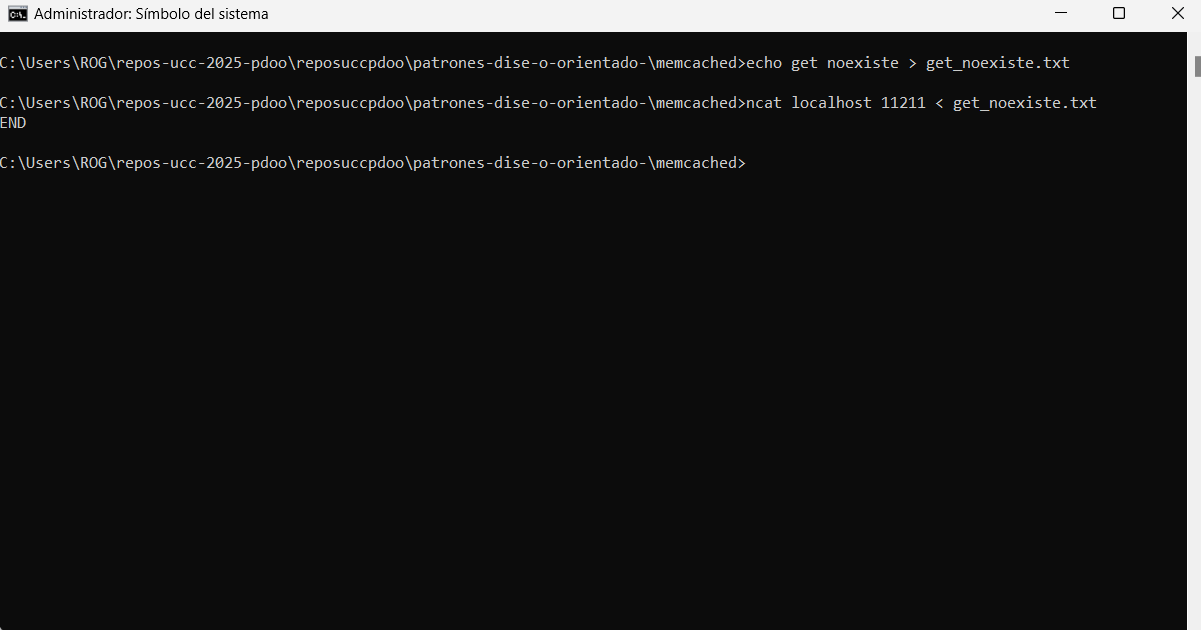
Prueba de Implementacion



1. **Comprobar comportamiento ante clave inexistente**

echo -e "get noexiste" | nc localhost 11211

Prueba de Implementacion

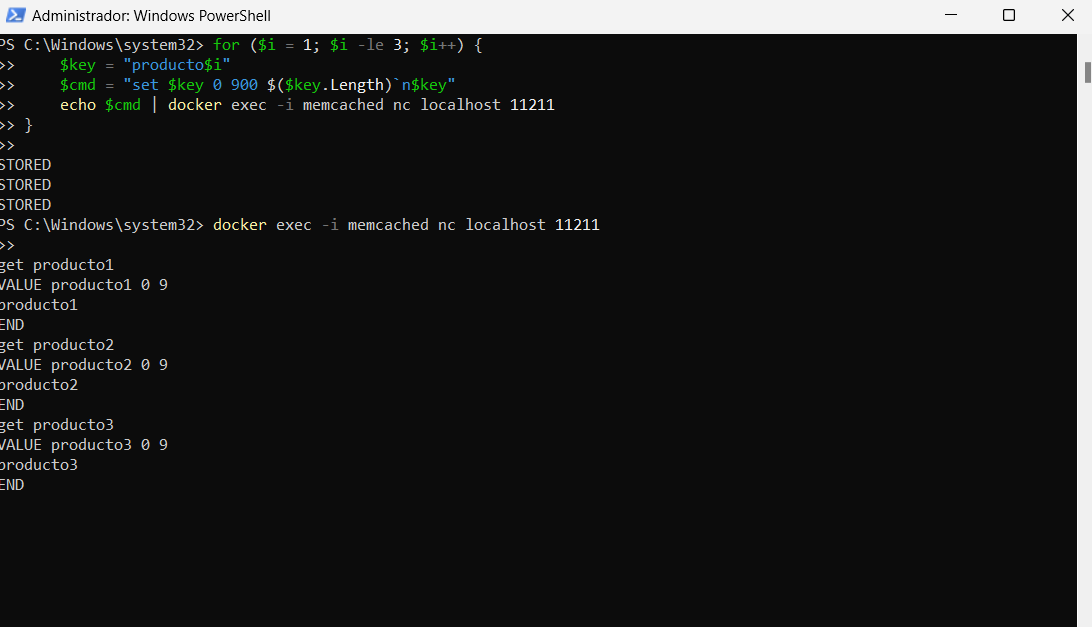


1. **Agregar múltiples claves**

for i in $(seq 1 3); do echo -e "set producto$i 0 900 8

producto$i" | nc localhost 11211; done

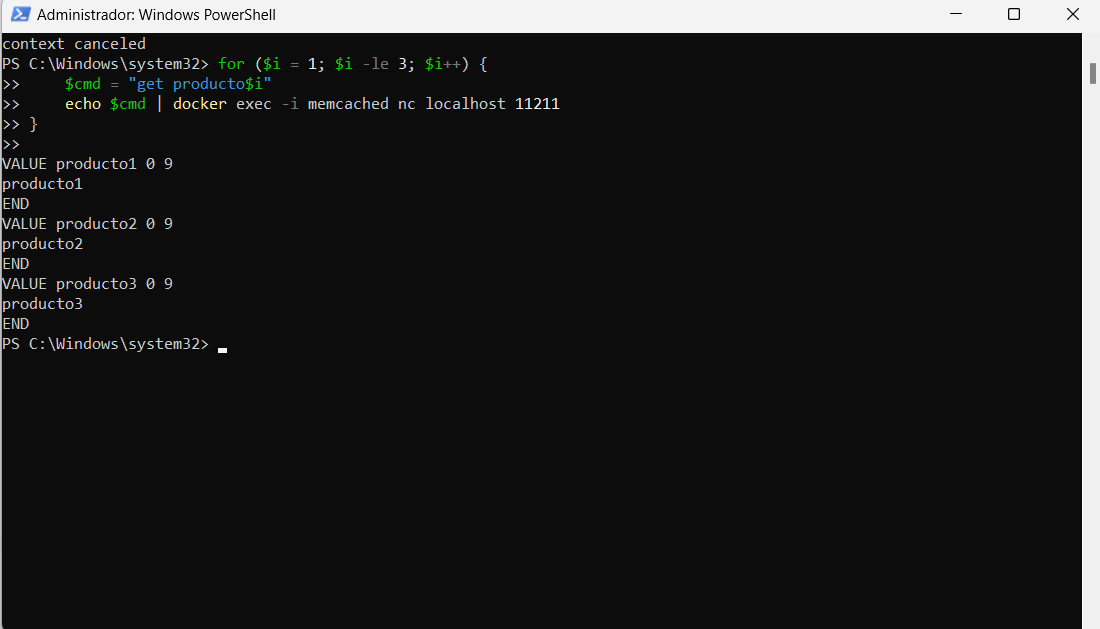
Prueba de Implementacion



1. **Verificar múltiples valores**

for i in $(seq 1 3); do echo -e "get producto$i" | nc localhost 11211; done

Prueba de Implementacion



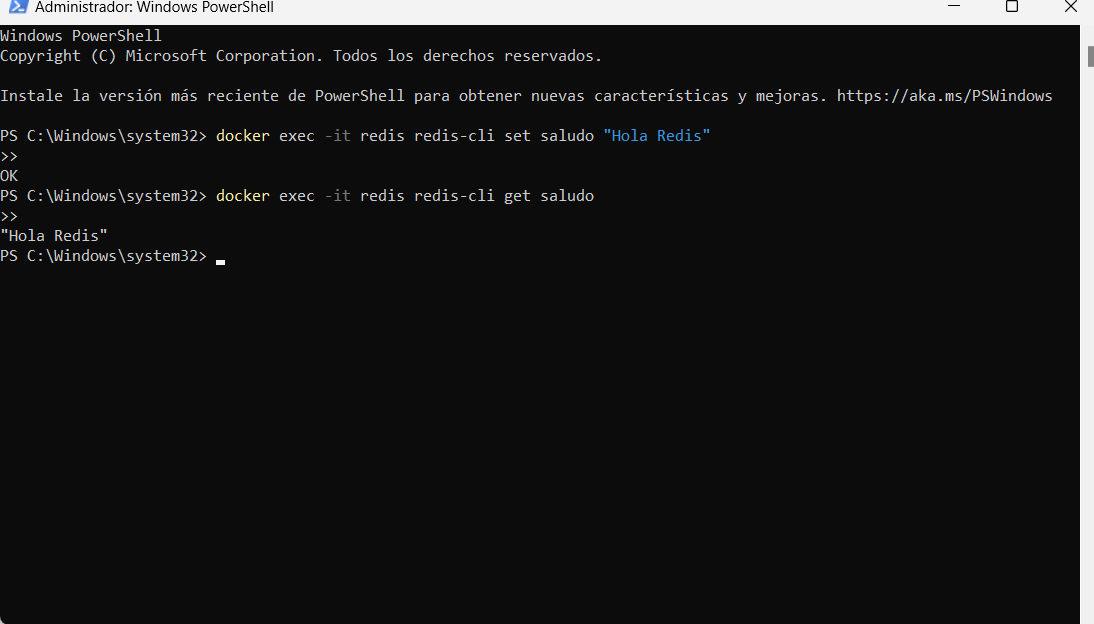
## Ejercicios con Redis

1. **Set y Get de una clave**

docker exec -it redis redis-cli set saludo "Hola Redis"

docker exec -it redis redis-cli get saludo

Prueba de Implementacion

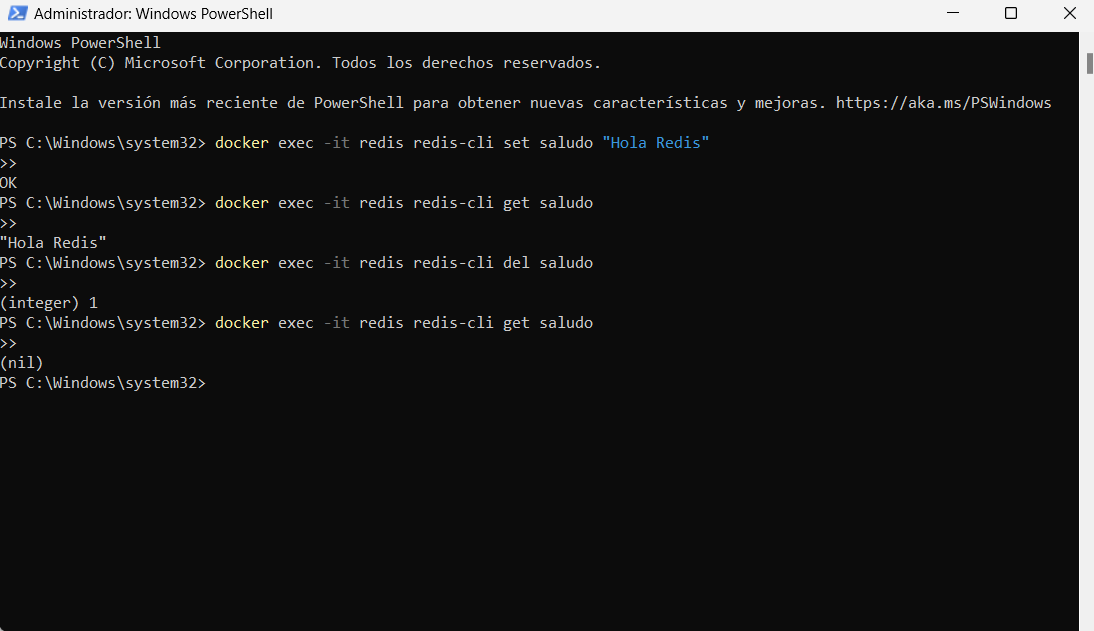


1. **Eliminar una clave y confirmar su eliminación**

docker exec -it redis redis-cli del saludo

docker exec -it redis redis-cli get saludo

Prueba de Implementacion

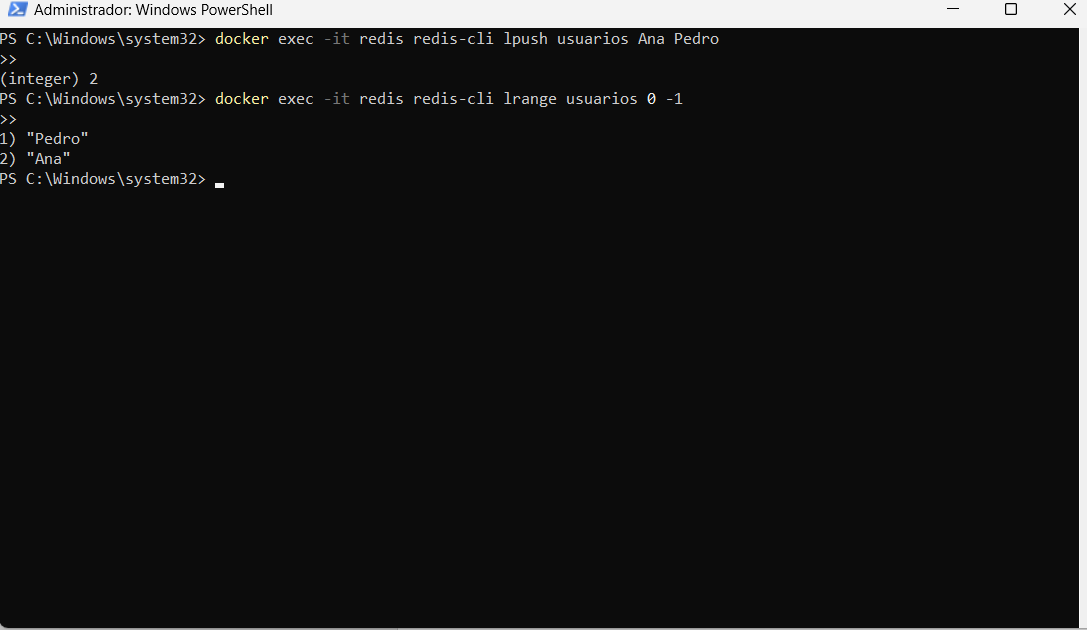


1. **Uso de listas**

docker exec -it redis redis-cli lpush usuarios Ana Pedro

docker exec -it redis redis-cli lrange usuarios 0 -1

Prueba de Implementacion



1. **Publicar y suscribirse a un canal**

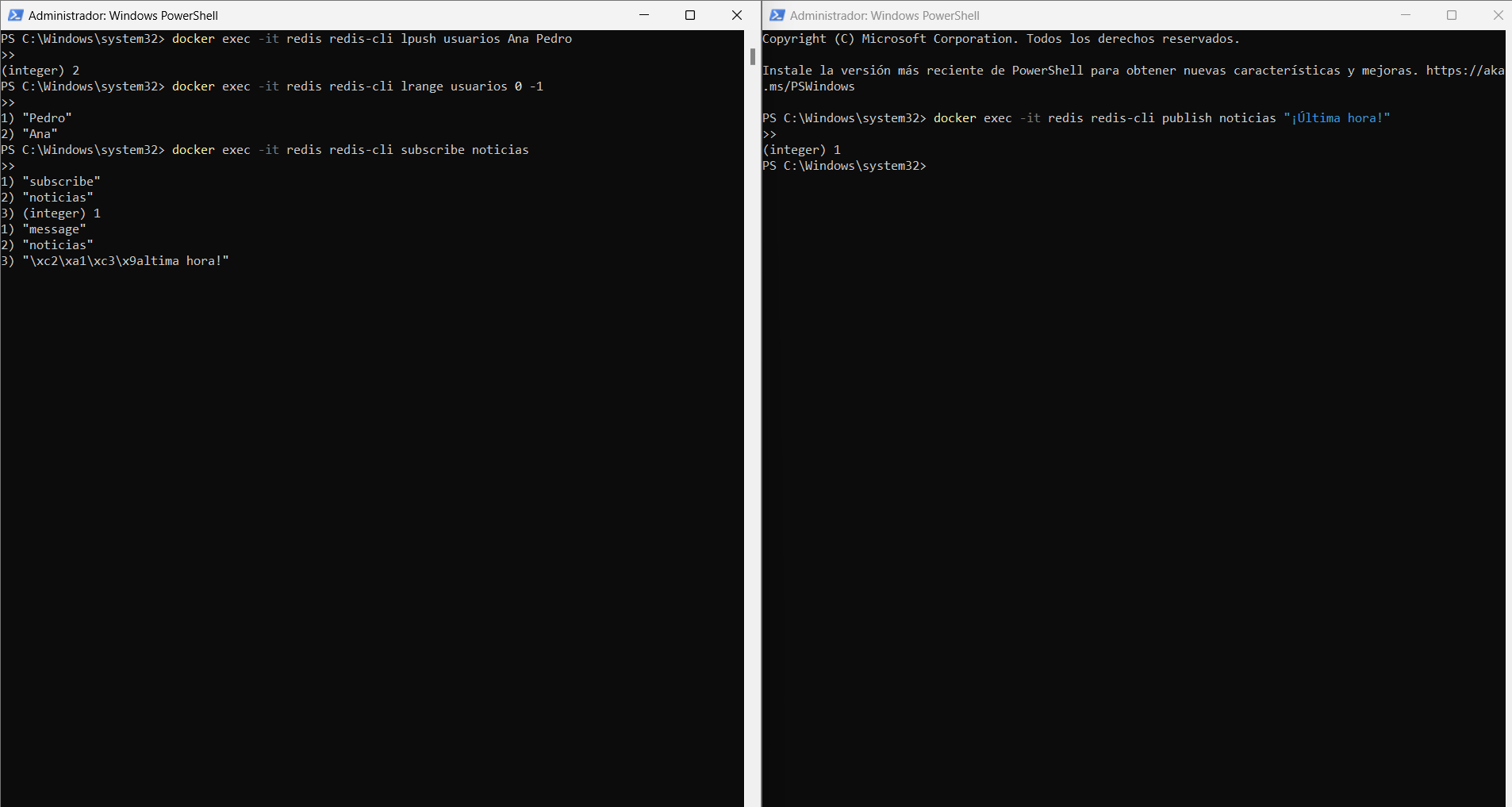
# En una terminal

redis-cli subscribe noticias

# En otra terminal

redis-cli publish noticias "¡Última hora!"

Prueba de Implementacion



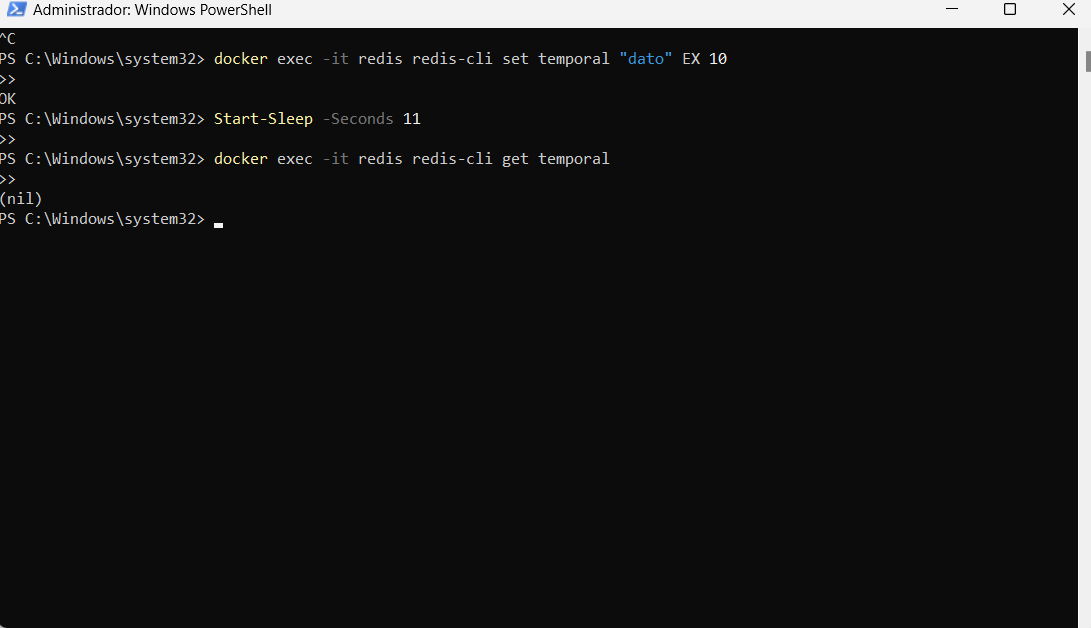
1. **Expirar una clave tras 10 segundos**

docker exec -it redis redis-cli set temporal "dato" EX 10

sleep 11

docker exec -it redis redis-cli get temporal

Prueba de Implementacion

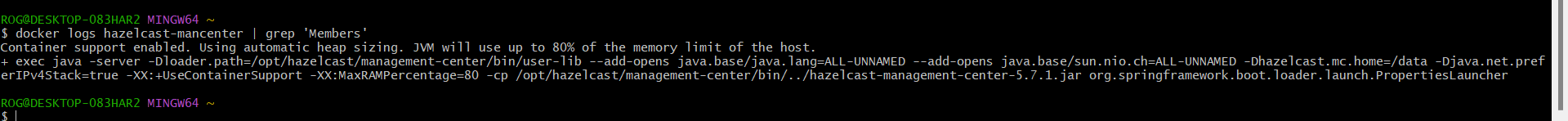


## Ejercicios con Hazelcast

1. **Verificar si el contenedor está en ejecución**

docker logs hazelcast | grep 'Members'

Prueba Implementacion



1. **Descargar cliente CLI para conectar vía REST (requiere curl)**

curl http://localhost:5701/hazelcast/rest/maps/sesiones -X PUT -d "usuario=Maria"

curl http://localhost:5701/hazelcast/rest/maps/sesiones/usuario

Prueba de Implementacion



1. **Insertar múltiples pares clave-valor**

for i in {1..3}; do curl -X PUT -d "valor$i" http://localhost:5701/hazelcast/rest/maps/datos/clave$i; done

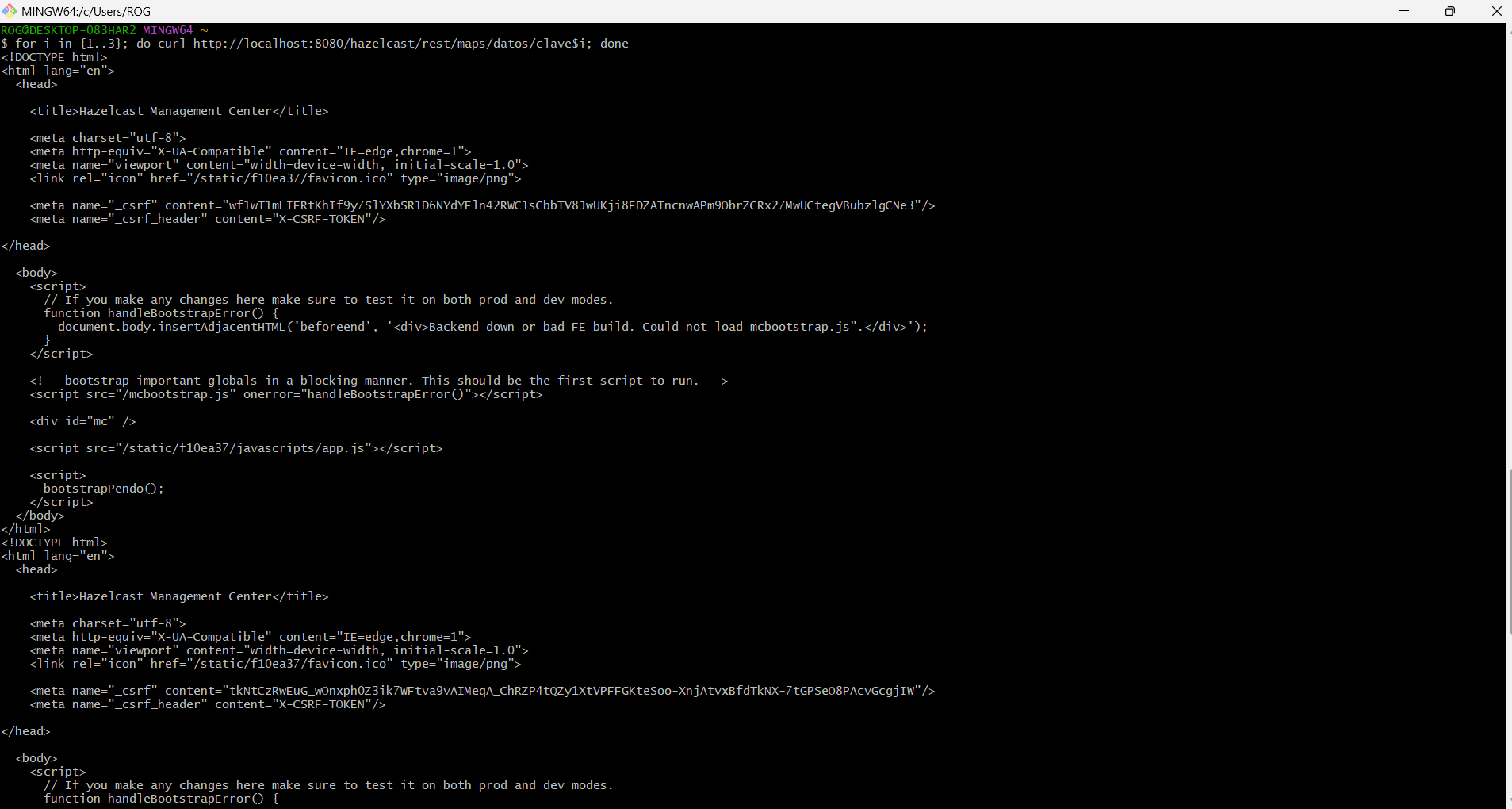
Prueba de Implementacion

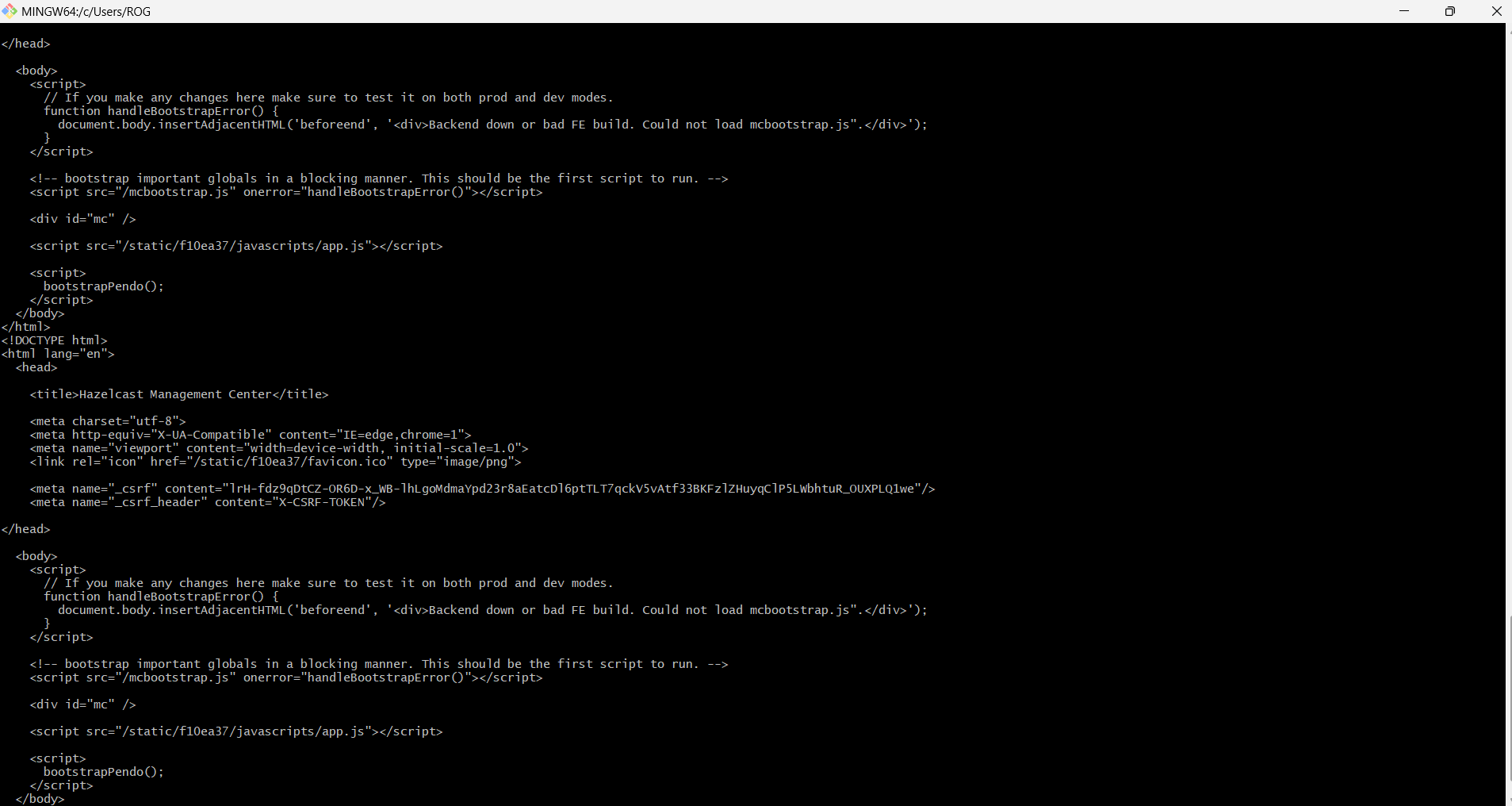


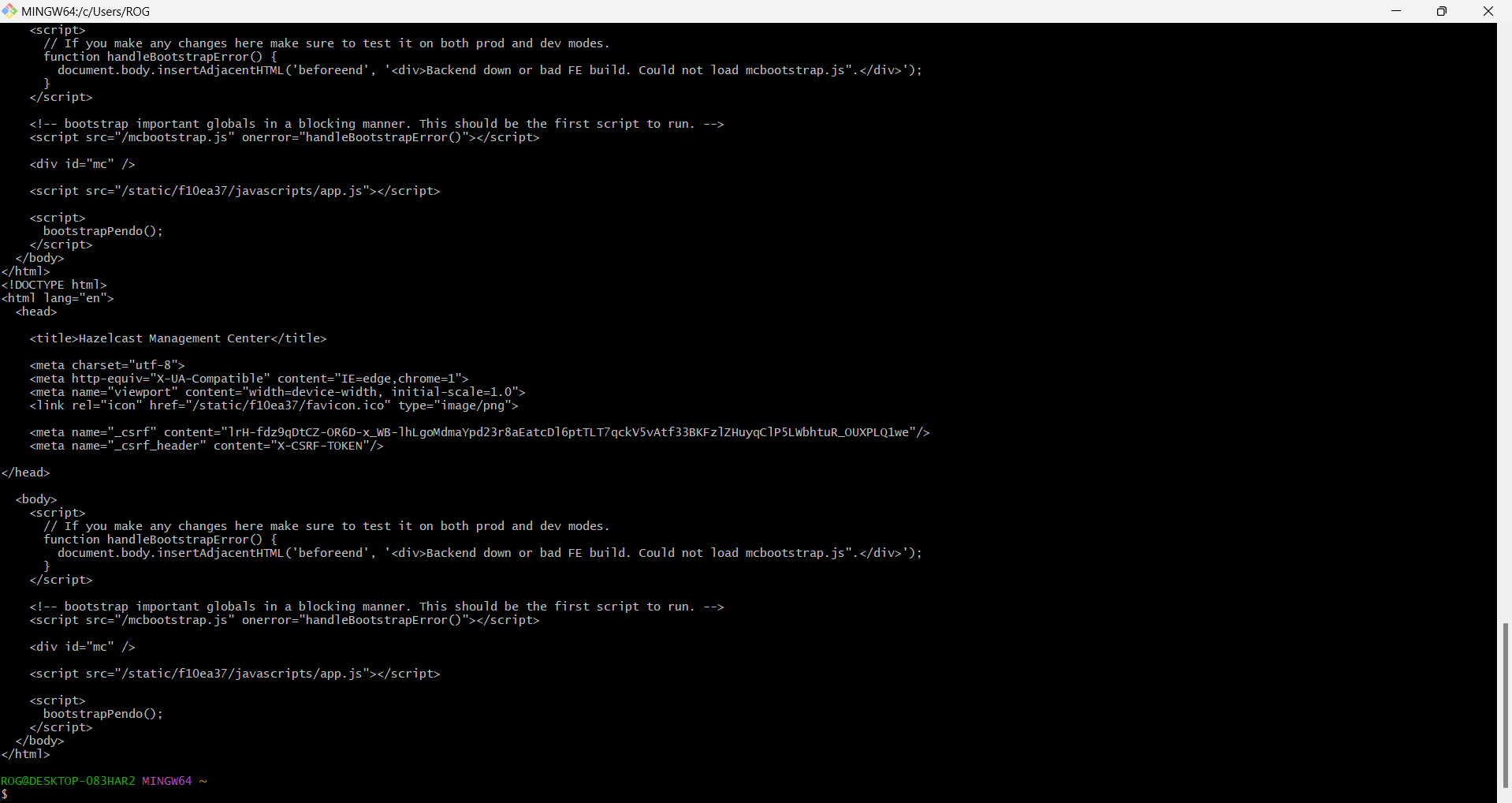
1. **Verificar consistencia recuperando claves**

for i in {1..3}; do curl http://localhost:5701/hazelcast/rest/maps/datos/clave$i; done

Prueba de Implementacion



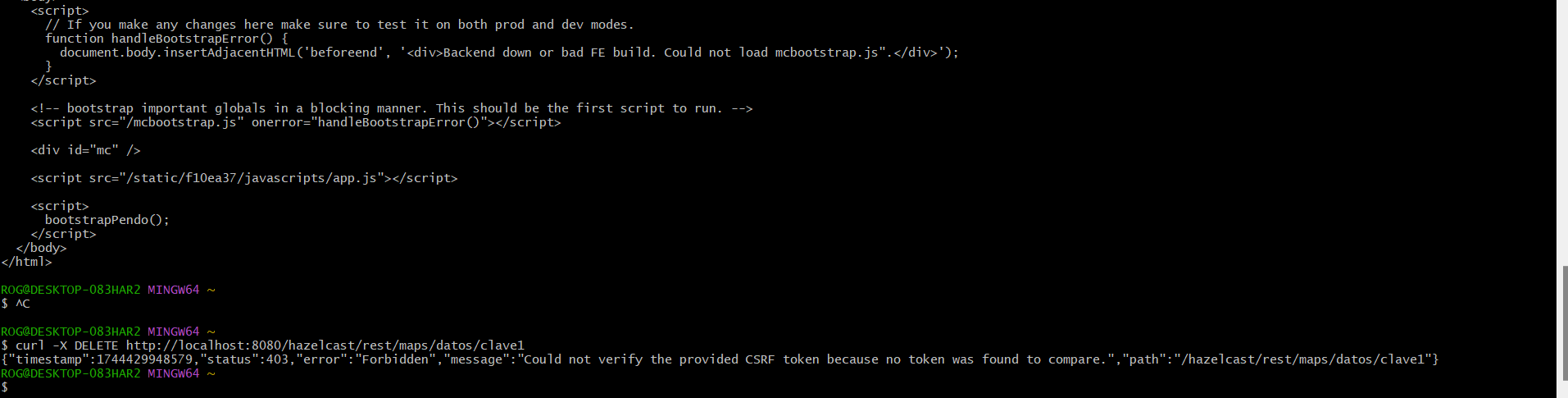




1. **Eliminar una clave**

curl -X DELETE http://localhost:5701/hazelcast/rest/maps/datos/clave1

Prueba de Implementacion



Link Repositorio

<https://github.com/ElCubaX/patrones-dise-o-orientado-/tree/main/memcached>