

Máster en Cloud Apps  
Desarrollo y despliegue de aplicaciones en la nube

Curso académico 2021/2022

Trabajo de Fin de Máster

**Your Race**

Autores: Rafael Gómez Olmedo,

Raquel Toscano Marchena

Tutor: Micael Gallego



**Contenido**

[Resumen](#_dsgy1lpaq26s)

[Introducción](#_mjcto5t97qqn)

[Objetivos](#_9fgqxwhrlfln)

[Alcance](#_wqz6a2ys2cn)

[Modelo del dominio](#_csr4k9mdb5ht)

[Requisitos](#_u7v8qk4pjact)

[Requisitos Funcionales](#_z6rkv9ovk8yt)

[Requisitos No Funcionales](#_mqdpr9oc6bcn)

[Asunciones](#_t5d3a85baavj)

[Fuera de Alcance](#_cwjcuksbkvdd)

[Casos de Uso](#_tojz1xf0qb3q)

[Actores](#_3xwzrty40kom)

[Casos de Uso](#_6glhely3vcsg)

[Prototipos de Interfaces](#_85xatx838jl1)

[Arquitectura](#_h6kah5x069i2)

[Implementación](#_cmzzlz1oofbo)

[Testing](#_xidsjxs5ncvp)

[Despliegue](#_2h7jqhaduny5)

[Conclusiones y trabajos futuros](#_j8agbyiqin1e)

[Bibliografía](#_th7980220rl7)

# 

# Resumen

Your Race is a scalable platform for managing entry assignment in highly demanded races.

<<Incluir agradecimientos>>

# Introducción

Inspirado en carreras populares con alta demanda:

* 101 Km de Ronda (<https://www.lalegion101.com/>)
* Maratón de Nueva York (<https://www.nyrr.org/tcsnycmarathon>)

El problema a resolver se inspira en lo expuesto aquí:

[1] [Cómo conseguir una plaza en los 101 kilómetros de Ronda](https://andaluciarunning.com/como-conseguir-una-plaza-en-los-101-kilometros-de-ronda/#:~:text=%2D%20Abre%20varias%20pesta%C3%B1as%20en%20el,el%20servidor%20del%20sitio%20web).

[2] [Así funciona la loteria](https://maratondeny.com/asi-funciona-la-loteria/)

Se trata de carreras populares, con muy alta demanda y que ofrecen un número de plazas limitado, normalmente de proporción bastante menor.

En el caso de la 101 de Ronda, tal y como se expone en el artículo anterior, las personas con interés suelen recurrir a trucos tales como abrir varias pestañas en el navegador con antelación suficiente para “garantizar que <<si se cae>> podrás saltar a la siguiente” o pedir colaboración a familiares y amigos.

<<PONER FOTO DE SERVIDOR CAÍDO?>>

[1] <https://andaluciarunning.com/como-conseguir-una-plaza-en-los-101-kilometros-de-ronda/#:~:text=%2D%20Abre%20varias%20pesta%C3%B1as%20en%20el,el%20servidor%20del%20sitio%20web>

[2] <https://maratondeny.com/asi-funciona-la-loteria/>

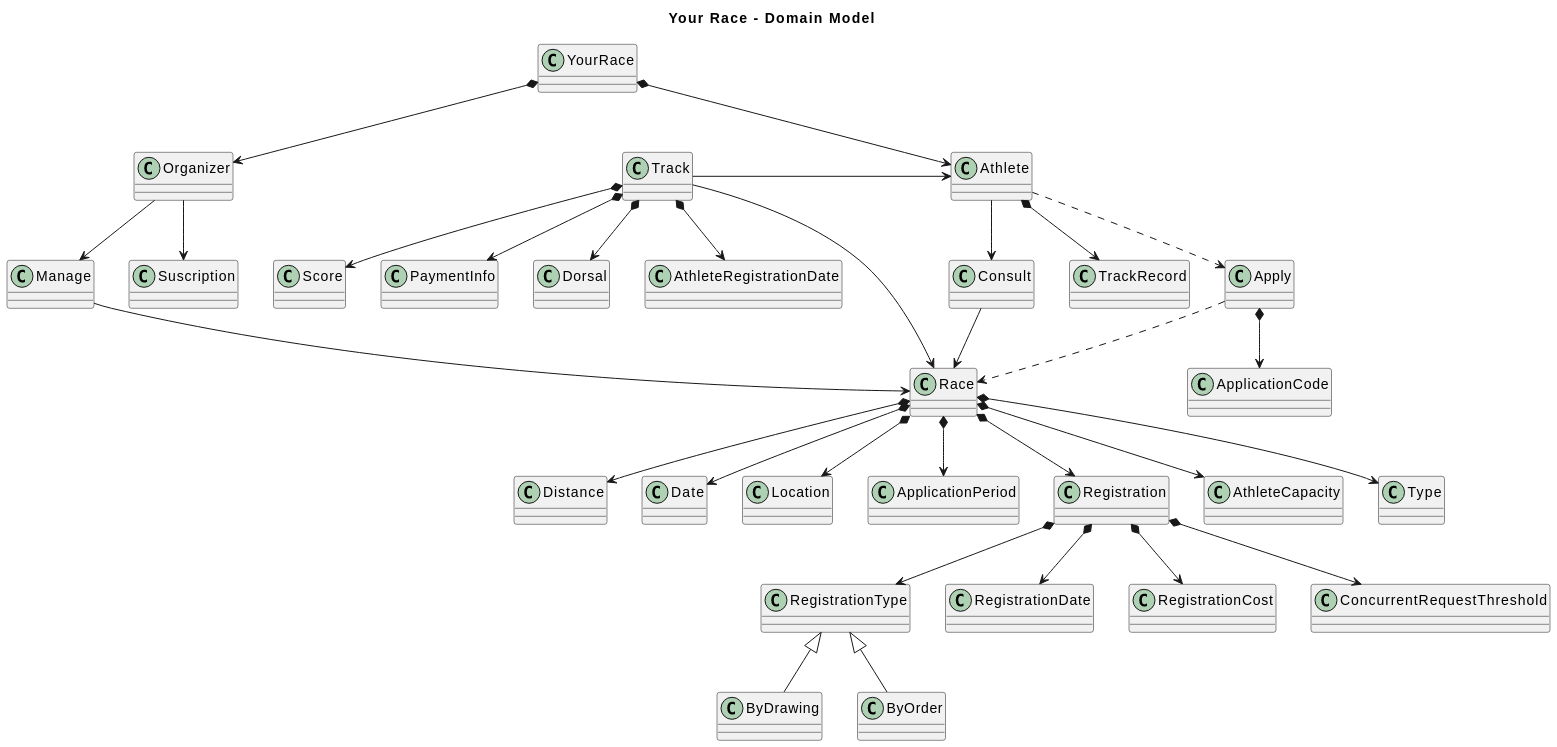
# Objetivos

El objetivo de your-race es ofrecer una plataforma escalable para gestionar la asignación de dorsales para carreras populares muy demandadas.

El principal objetivo es asegurar la continuidad del servicio en todo momento, especialmente a la hora exacta en la que se espera la avalancha de conexiones.

# Alcance

## Modelo del dominio



## Requisitos

### Requisitos Funcionales

Ofrecer un sistema para gestionar el registro de atletas en carreras de alta demanda.

Existen 2 tipos de roles: atletas y organizadores.

Organizadores:

* Deben crear una suscripción a nuestra plataforma.
* Gestionan las carreras.

Atletas:

* Pueden preinscribirse a una carrera determinada.
* Como resultado de la preinscripción, obtienen un ApplicationCode.
* Cuentan con un palmarés de marcas personales (TrackRecord).
* Consultan la información de carreras disponibles (hayan participado o no).

Carreras:

* Cuentan con la siguiente información: Distancia, Fecha, Localización, Score, Periodo de preinscripción (ApplicationPeriod), Tipo (carrera, bicicleta)
* Cuentan también con un Registro.

Registro:

* Es el mecanismo por el que un atleta preinscrito a una carrera obtiene dorsal (y entrada) a la misma.
* Tiene información de pago (tasas, estado, etc), dorsal (número).
* Se produce en una fecha determinada.
* Existen dos tipos de registro:
  + Por sorteo.
  + Por orden de llegada.

### Requisitos No Funcionales

* Se debe asegurar en todo momento la continuidad del servicio. Aquí está nuestro negocio y no podemos fallar en esto.
* High performance and scalability.
* Debe soportar una carga de demanda de 25.000 peticiones concurrentes en un momento determinado, ya que en el peor de los casos se estima que todos los atletas con preinscripción van a acceder a la misma hora.
* Es posible prever en qué momento se va a producir esta avalancha de demanda. PROGRAMACIÓN AUTOMÁTICA DE LA HORA EN LA QUE SE PUEDE PREPARAR EL SISTEMA PARA LA INFRA. CONSULTAR A MICA. Algo así como monitorizar las fechas de registro en las carreras programadas para anticiparse a la hora en la que comienzan.
* Debe existir tener unas tablas de escalabilidad y facturación dimensionadas con los resultados de peformance tests. Que de ahí salga tanto el nº de pods como la facturación del servicio (tarifas,...).
* Es posible que aunque tengamos previsto el pico de 25.000, finalmente sea algo mayor (30.000) ya que los atletas pueden haber solicitado ayuda a familiares y amigos.
* Cada vez que haya una nueva feature o de **forma automática diariamente por las noches**, se deberá comprobar que los tiempos no se ven perjudicados a través de los correspondientes pruebas de performance. En caso contrario hay que generar una alarma.
* Un factor a determinar será el decidir cuándo se va a producir el desescalado de la aplicación, debiendo ajustarse lo máximo posible.
* Authorization and Authentication

### Asunciones

<<Completar si procede>>

### Fuera de Alcance

<<Completar si procede>>

## Casos de Uso

### Actores

### Casos de Uso

### Prototipos de Interfaces

# Arquitectura

# Implementación

# 

# Testing

TDD y tipos de test incluídos.

# Despliegue

# Observabilidad y Monitorización

# Performance Testing

++ Bare metal utilizado.

++ Tests y conclusiones por histórico: cosas que hemos ido añadiendo y cómo ha ido mejorando.

Y pruebas que hemos hecho y no han dado ningún resultado.

1º Tramo: Tests exploratorios. Búsqueda de punto óptimo.

2º Tramo: Muerte de BBDD en peticiones concurrentes.

3º Tramo: Circuit Breaker.

4º Tramo: RabbitMQ.

Consultar ejecución detallada en el anexo YourRaceTestingScenarios[1].

[1]<https://docs.google.com/spreadsheets/d/1K2KCRoR6Kmkq3UN-WFXWZY6JZzejWN9V1HZ-6EVJA_Q/edit?usp=sharing>

ECONNREFUSED

ETIMEDOUT

01 - Created 201

202 - Accepted

200 - ClientAbortException Error

400 - AthleteAlreadyRegisteredToRace

Error 400 - RaceFullCapacityException

Error 500 - Server - CannotCreateTransactionException

Error 500 - Server - DataIntegrityViolationException

Error 503 - client - Service Unavailable

Error 504 - client - Gateway Timeout

DatabaseTracks registrations

## 

## Testing Exploratorio.

1000 peticiones por segundo, 30 segundo

## Búsqueda del punto óptimo

## Mejoras capa de negocio

## Mejoras datos de entrada

## 

## Muerte de BBDD en peticiones concurrentes

Feature Toggle:

Use RabbitMQ producer = DISABLED

ERRORES CON -> MUERTE DEL CLUSTER

TC 121. Prueba 1 tras arrancar el clúster.

Cambiado el código para devolver 200 en lugar de 201. Comienzo pruebas wave4. Empiezo sin RabbitMQ -> Rabbit ha encolado peticiones. Aclarar esto -> Hay que poner Rabbit en el manifiesto a false para que todos los pods lo tengan a falso. Repito pero levantando los pods con configuración a false

->76% de registros y errores CONN

TC 122. Prueba 2 tras arrancar el clúster.

Mismas condiciones.

TC 123. Prueba 3 tras arrancar el clúster. -> Los resultados ahora se normalizan.

No se han normalizado aún, pero todos mejoran. COMENTAR ERRORES CON-> NO SE VAN HASTA MATAR CLUSTER.

TC 124. Prueba 4 tras arrancar el clúster. -> La prueba de que los resultados ahora se normalizan.

TC 125. Prueba 5 tras arrancar el clúster. -> La prueba de que los resultados ahora se normalizan.

El portforward de Grafana revienta

TC 126. Parada y re-arrancado de clúster -> Vuelven los errores.

## Muerte de BBDD en peticiones concurrentes

Pruebas anteriores: 121 a 126

Errores:

{exception="AthleteAlreadyRegisteredToRace", status="400"}

Max: 91.1

{exception="CannotCreateTransactionException", status="500"}

Max: 91.0

{exception="ClientAbortException", status="200"}

Max: 0.269

{exception="DataIntegrityViolationException", status="500"}

Max: 78.1

{exception="None", status="200"}

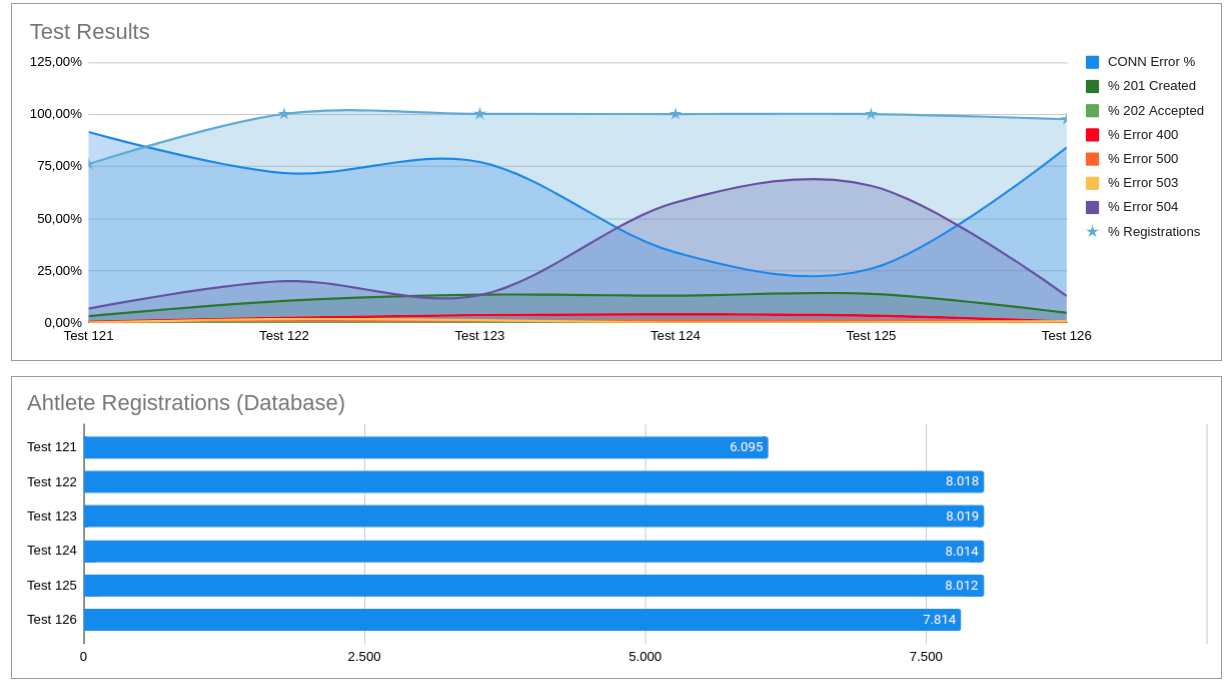
Max: 0.0950

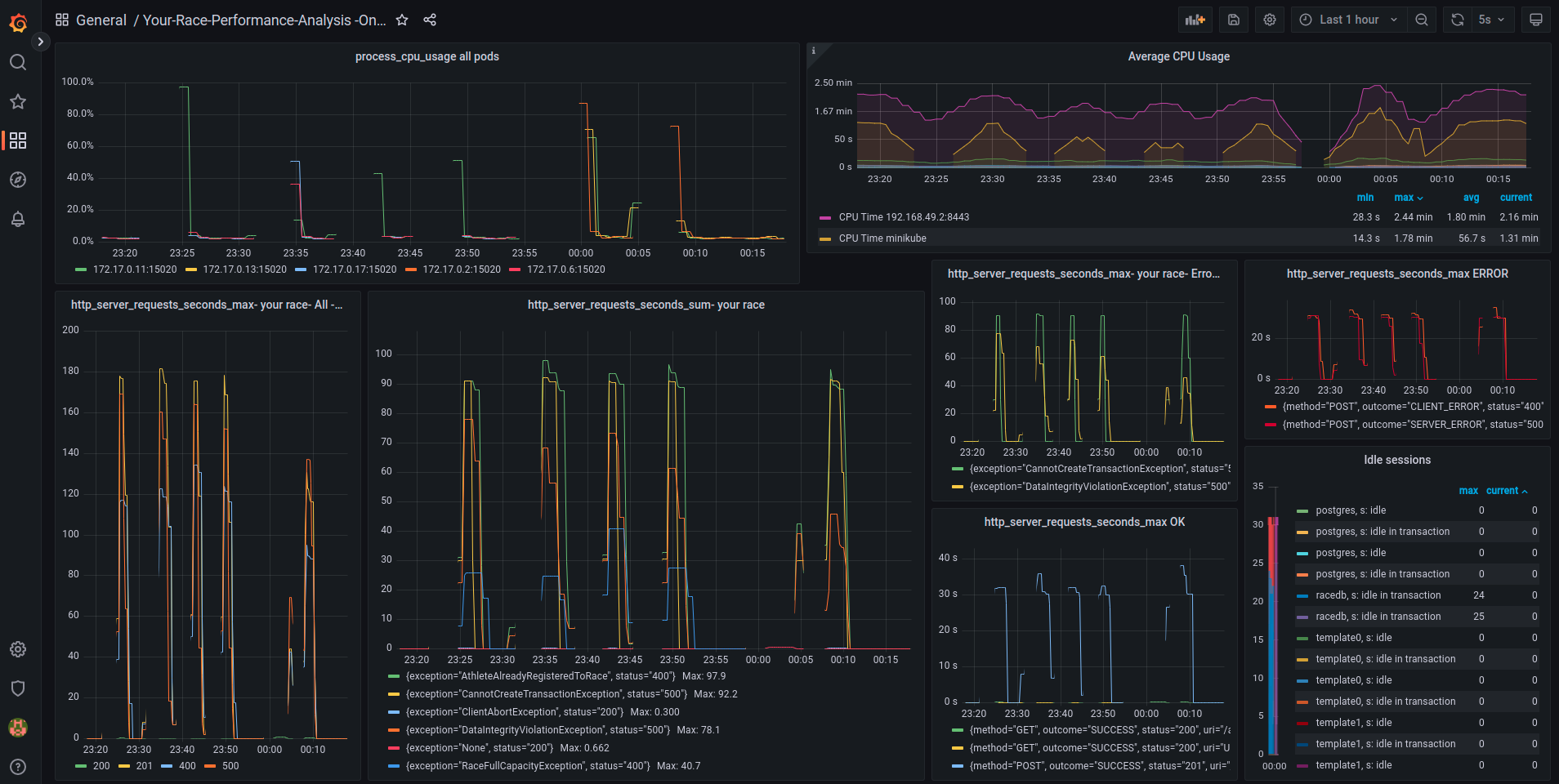
{exception="RaceFullCapacityException", status="400"}

Max: 25.9

Se registran más atletas que la capacidad de la carrera

El Feature Toggle hay que ponerlo en el deployment para que lo cojan todas las réplicas. Si no -> hacerlo por pod… cómo?





Prueba Mongo

## 

## RabbitMQ

## 

TC 127. Prueba 1 tras arrancar el clúster.

TC 128. Prueba 2 tras arrancar el clúster.

Mismas condiciones.

TC 129. Prueba 3 tras arrancar el clúster. -> Los resultados ahora se normalizan.

TC 130. Prueba 4 tras arrancar el clúster. -> La prueba de que los resultados ahora se normalizan.

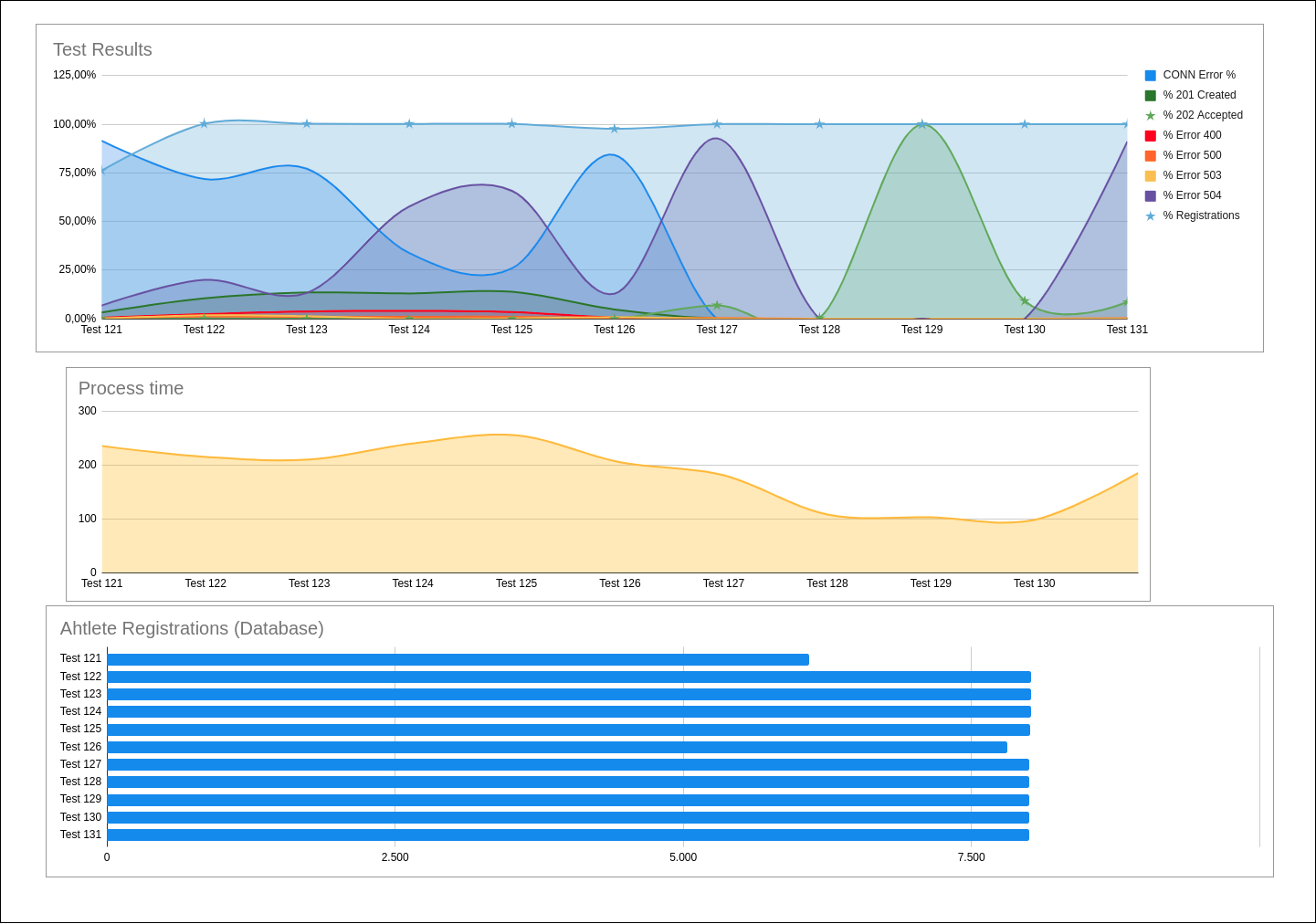
TC 131. Parada y re-arrancado de clúster -> Vuelven los errores.

Desaparición de errores 400

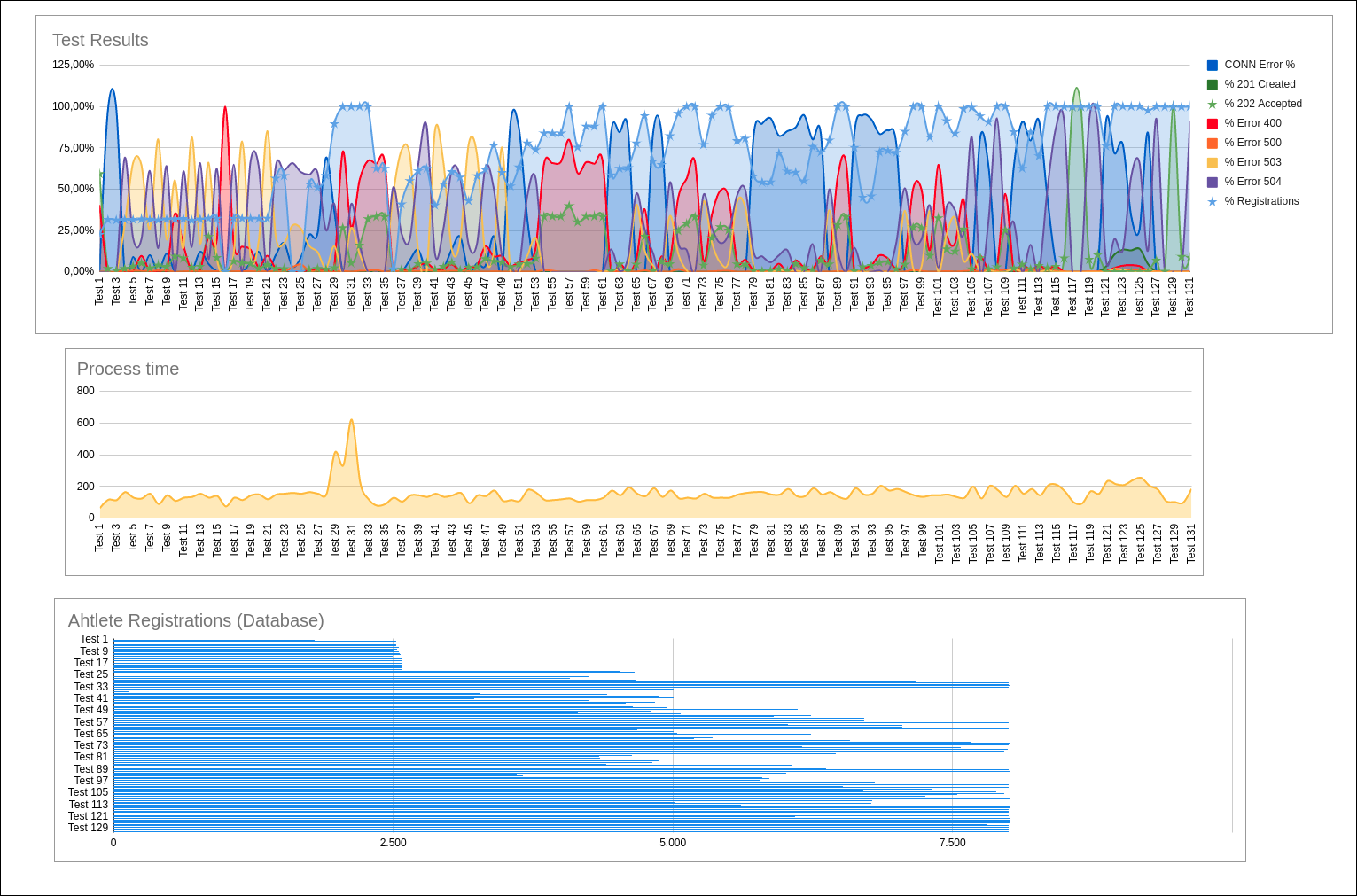
## Arranque de clúster (tras creación o parada)

PRUEBA DE ARRANCADO Y ESTABILIZACIÓN CON RABBIT

Pruebas anteriores.



De un vistazo:



# Conclusiones y trabajos futuros

# Bibliografía