**Testing Report**

## **Student 2**



## **Integrante**

* Noya Cano, Lucía: lucnoycan@alum.us.es

## **Repositorio**

<https://github.com/DP2-C1-035/Acme-ANS-C2.git>

02/07/2025

## **ÍNDICE**

Contenido

[Student 2 1](#_Toc1667749455)

[Integrante 1](#_Toc627264897)

[Repositorio 1](#_Toc1453641734)

[ÍNDICE 1](#_Toc751356314)

[Resumen ejecutivo 2](#_Toc491980016)

[Tabla de versiones 2](#_Toc1313362818)

[Introducción 3](#_Toc2061329273)

[Cobertura del código 4](#_Toc1121665747)

[Casos de prueba de Booking 5](#_Toc709093700)

[Casos de prueba de BookingRecord 7](#_Toc766590040)

[Casos de prueba de Passenger 8](#_Toc316051828)

[Performance testing (Ordenador 1) 10](#_Toc273621807)

[Performance testing después de añadir índices (Ordenador 2) 12](#_Toc555719400)

[Contraste de 95% de confianza comparación con otro ordenador 13](#_Toc1547568581)

[Conclusión 14](#_Toc1341116051)

[Bibliografía 14](#_Toc1899185419)

# Resumen ejecutivo

En este documento se recoge toda la información relevante del testing llevado acabo por el Student2 en el proyecto DP2-C2-035, detallando el plan seguido a la hora de realizar los casos de prueba. También se detalla el rendimiento obtenido al repetir los tests en otra máquina distinta y una comparativa estadística sobre el rendimiento en cada caso.

# Tabla de versiones

|  |  |  |
| --- | --- | --- |
| **Versión** | **Cambio** | **Fecha** |
| 1.0.0 | Inicio del documento | 02/07/2025 |
| 1.1.0 | Añadida conclusión | 02/07/2025 |

# Introducción

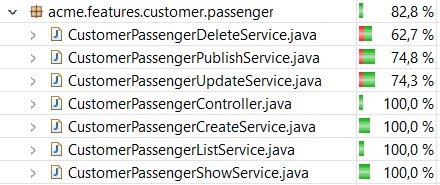
En el presente documento se describe la metodología de testing que ha usado el Student2 durante el entregable C2.

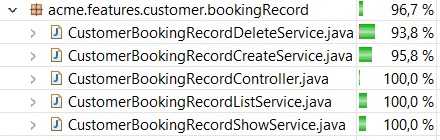
El documento contiene un resumen ejecutivo que proporciona una visión clara de los puntos más importantes del documento, así como la información necesaria para entenderlo. Además, el documento cuenta con una lista de versiones como se puede comprobar en la tabla de versiones.

En cuanto al contenido de este, está dividido en varios apartados que se corresponden con la cobertura total lograda en el código, casos de prueba para las distintas features implementadas indicando el resultado esperado de cada caso de prueba, tres análisis de rendimiento del código realizados en diferentes circunstancias y, por último, una comparativa sobre los valores de rendimiento obtenidos con anterioridad según el caso en el que se ha llevado a cabo.

# Cobertura del código







# Casos de prueba de Booking

A continuación se muestra una tabla con los valores probados para cada atributo en los formularios de creación y actualización. Para todos los atributos se probó dejar el campo vacío y comprobar que fuese inválido.

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Valor válido** | **Valor inválido** |
| locatorCode | Longitud de 6, 7 y 8 caracteres | Longitud de 1, 5 y 9 caracteres  locatorCode ya existente como por ejemplo JQ1V0Z |
| travelClass | ECONOMY y BUSINESS |  |
| price | EUR 0.01  EUR 99,999.99  EUR 50.00 | XXX  XXX 21  EUR  EUR -0.01  EUR 100,000.01 |
| lastNibble | 1234 | 1  12345 |

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción de acción realizada** | **Resultado** |
| customer/booking/list | Listar bookings de un customer logeado como customer | Listado de bookings de un customer |
| customer/booking/list | Listar bookings de un customer sin logear | Excepción unauthorise |
| customer/booking/list | Listar bookings de un customer logeado como administrator | Excepción unauthorise |
| customer/booking/show | Mostrar un booking de un customer estando logeado como ese customer | Formulario show del booking del customer |
| customer/booking/show | Mostrar un booking de un customer sin estar logeado | Excepción unauthorise |
| customer/booking/show | Mostrar un booking de un customer estando logeado como administrator | Excepción unauthorise |
| customer/booking/show | Mostrar un booking de un customer estando logeado como otro customer | Excepción unauthorise |
| customer/booking/show | Mostrar un booking nulo | Excepción unauthorise |
| customer/booking/create | Crear un booking con algún campo inválido pero legal | Error en aquellos campos inválidos |
| customer/booking/create | Acceder a la feature sin estar logeado | Excepción unauthorise |
| customer/booking/create | Acceder a la feature logeado como administrator | Excepción unauthorise |
| customer/booking/update | Actualizar un booking válido con algún campo inválido pero legal | Error en aquellos campos inválidos |
| customer/booking/update | Actualizar un booking con un flight no mostrado en el listado en el desplegable | Excepción unauthorise |
| customer/booking/update | Acceder a la feature sin estar logeado | Excepción unauthorise |
| customer/booking/update | Acceder a la feature logeado como administrator | Excepción unauthorise |
| customer/booking/update | Actualizar un booking que no está en modo borrador | Excepción unauthorise |
| customer/booking/update | Actualizar un booking válido con un flight que ya ha despegado | Error en el campo flight indicando que no es un error 500 |
| customer/booking/publish | Publicar un booking válido de un customer estando logeado como ese customer | Se publica el booking |
| customer/booking/publish | Publicar un booking nulo | Excepción unauthorise |
| customer/booking/publish | Publicar un booking ya publicado de un customer estando logeado como ese customer | Excepción unauthorise |
| customer/booking/publish | Publicar un booking válido de un customer estando logeado como otro customer | Excepción unauthorise |
| customer/booking/publish | Publicar un booking sin pasajeros asociados | Error indicando que la booking no tiene pasajeros asociados |
| customer/booking/publish | Publicar un booking sin atributo lastNibble | Error indicando que no se puede publicar un booking sin lastNibble |
| customer/booking/publish | Publicar un booking de un customer sin estar logeado | Excepción unauthorise |
| customer/booking/publish | Publish un booking de un customer estando logeado como administrator | Excepción unauthorise |

# Casos de prueba de BookingRecord

Para esta feature solo hay un campo disponible para probar. Es un desplegable para seleccionar el passenger asociado al booking. Ese campo se prueba con pasajeros existentes o no existentes.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción de acción realizada** | **Resultado** |
| customer/bookingRecord  /list | Listar pasajeros asociados a un booking de un customer logeado como ese customer | Listado de pasajeros asociados al booking |
| customer/bookingRecord  /list | Listar pasajeros asociados a un booking null | Excepción unauthorise |
| customer/bookingRecord  /list | Listar pasajeros asociados a un booking de un customer logeado como otro customer | Excepción unauthorise |
| customer/bookingRecord  /list | Listar pasajeros asociados a un booking de un customer sin estar logeado | Excepción unauthorise |
| customer/bookingRecord  /list | Listar pasajeros asociados a un booking de un customer logeado como administrator | Excepción unauthorise |
| customer/bookingRecord  /show | Mostrar un pasajero asociado a un booking de un customer logeado como ese customer | Mostrar formulario de pasajero en pantalla |
| customer/bookingRecord  /show | Mostrar un pasajero asociado a un booking de un customer logeado como ese customer | Mostrar formulario de pasajero asociado en pantalla |
| customer/bookingRecord  /show | Mostrar un pasajero asociado a un bookinRecord nulo | Excepción unauthorise |
| customer/bookingRecord  /show | Mostrar un pasajero asociado a un booking de un customer logeado como otro customer | Excepción unauthorise |
| customer/bookingRecord  /show | Mostrar un pasajero asociado a un booking de un customer logeado como administrator | Excepción unauthorise |
| customer/bookingRecord  /show | Mostrar un pasajero asociado a un booking de un customer sin estar logeado | Excepción unauthorise |
| customer/bookingRecord  /create | Asociar un pasajero válido a un booking no publicado de un customer logeado como ese customer | Asociar pasajero a la booking |
| customer/bookingRecord  /create | Asociar un pasajero inválido pero legal a un booking no publicado de un customer logeado como ese customer | Se muestra un error en pantalla |
| customer/bookingRecord  /create | Asociar un pasajero a un booking nulo | Excepción unauthorise |
| customer/bookingRecord  /create | Asociar un pasajero a un booking publicado | Excepción unauthorise |
| customer/bookingRecord  /create | Asociar un pasajero a un booking no publicado de un customer logeado como otro customer | Excepción unauthorise |
| customer/bookingRecord  /create | Asociar un pasajero no mostrado en el desplegable a un booking no publicado de un customer logeado como ese customer | Excepción unauthorise |

# Casos de prueba de Passenger

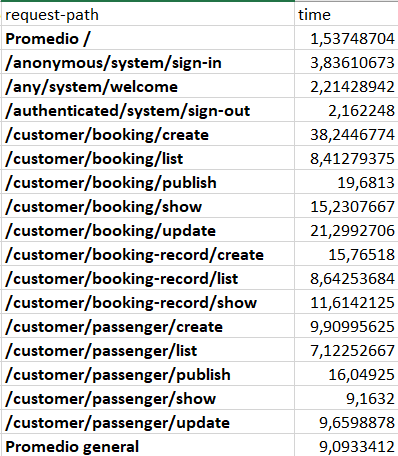
A continuación se muestra una tabla con los valores probados para cada atributo en los formularios de creación y actualización. Para todos los atributos se probó dejar el campo vacío y comprobar que fuese inválido, menos en el caso del atributo specialNeeds que puede ser nulo.

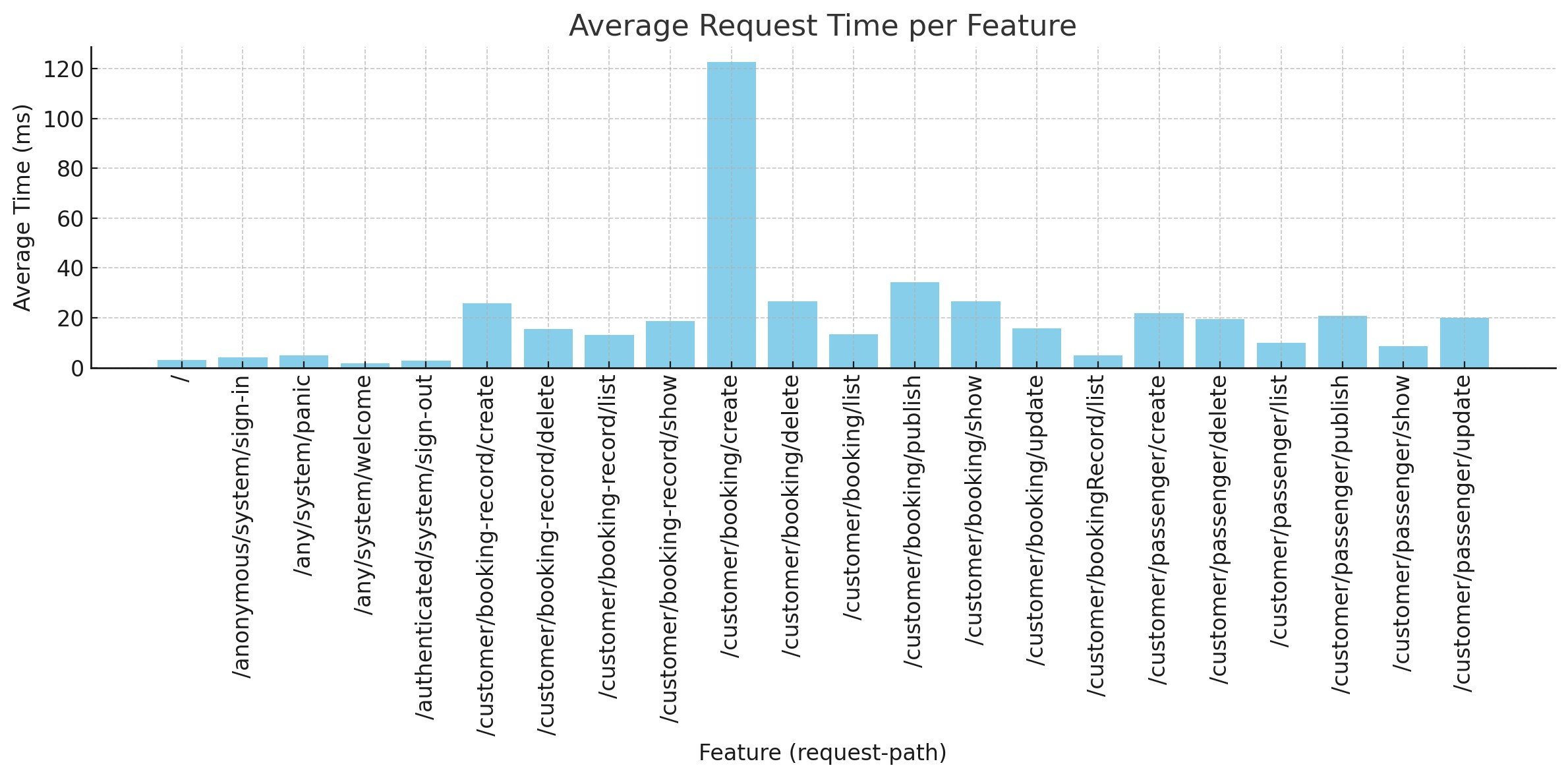
|  |  |  |
| --- | --- | --- |
| **Atributo** | **Valor válido** | **Valor inválido** |
| fullName | Longitud de 1, 2, 254 y 255 caracteres  Caracteres especiales como chino o árabe  Injections SQL | Longitud de 256 caracteres |
| email | a@b.e  a@b.es  a@b.esx  [lorem-ipsum@dolor.sit.amet.consectetur.adipiscing.elit.sed.do.eiusmod.tempor.incididunt.ut.labore.et.dolore.magna.aliqua.Ut.enim.ad.minim.veniam.quis.nostrud.exercitation.ullamco.laboris.nisi.ut.aliquip.ex.ea.commodo.consequat.duis.aute.irure.doloris.edu](mailto:lorem-ipsum@dolor.sit.amet.consectetur.adipiscing.elit.sed.do.eiusmod.tempor.incididunt.ut.labore.et.dolore.magna.aliqua.Ut.enim.ad.minim.veniam.quis.nostrud.exercitation.ullamco.laboris.nisi.ut.aliquip.ex.ea.commodo.consequat.duis.aute.irure.doloris.edu)  lorem-ipsum@dolor.sit.amet.consectetur.adipiscing.elit.sed.do.eiusmod.tempor.incididunt.ut.labore.et.dolore.magna.aliqua.Ut.enim.ad.minim.veniam.quis.nostrud.exercitation.ullamco.laboris.nisi.ut.aliquip.ex.ea.commodo.consequat.duis.aute.irure.dolor.es.org | user:password@mail.com  lorem-ipsum@dolor.sit.amet.consectetur.adipiscing.elit.sed.do.eiusmod.tempor.incididunt.ut.labore.et.dolore.magna.aliqua.Ut.enim.ad.minim.veniam.quis.nostrud.exercitation.ullamco.laboris.nisi.ut.aliquip.ex.ea.commodo.consequat.duis.aute.irure.dolor.est.com |
| passportNumber | Longitud de 9 caracteres | Longitud de 1,5 y 10 caracteres |
| specialNeeds | Longitud de 1,2 y 50 carcteres | Longitud de 51 caracteres |

|  |  |  |
| --- | --- | --- |
| **Feature** | **Descripción de acción realizada** | **Resultado** |
| customer/passenger/list | Listar pasajeros asociados a un customer logeado como ese customer | Listado de pasajeros del customer |
| customer/passenger/list | Listar pasajeros asociados a un customer logeado como ese administrator | Excepción unauthorise |
| customer/passenger/list | Listar pasajeros asociados a un customer sin estar logeado | Excepción unauthorise |
| customer/passenger/show | Mostrar pasajero asociado a un customer estando logeado como ese customer | Mostrar formulario de pasajero en panatalla |
| customer/passenger/show | Mostrar pasajero nulo | Excepción unauthorise |
| customer/passenger/show | Mostrar pasajero asociado a un customer estando logeado como otro customer | Excepción unauthorise |
| customer/passenger/show | Mostrar pasajero asociado a un customer estando logeado como administrator | Excepción unauthorise |
| customer/passenger/show | Mostrar pasajero asociado a un customer sin estar logeado | Excepción unauthorise |
| customer/passenger/create | Crear pasajero válido logeado como customer | Se crea un pasajero con los atributos del formulario |
| customer/passenger/create | Crear pasajero con algunos camos vacios logeado como customer | Error en todos los campos que no pueden ser nulos |
| customer/passenger/create | Crear pasajero logeado como administrator | Excepción unauthorise |
| customer/passenger/create | Crear pasajero sin estar logeado | Excepción unauthorise |
| customer/passenger/update | Actualizar pasajero válido asociado a un customer logeado como ese customer | Se actualiza el pasajero con los atributos del formulario |
| customer/passenger/update | Actualizar pasajero nulo | Excepción unauthorise |
| customer/passenger/update | Actualizar pasajero publicado | Excepción unauthorise |
| customer/passenger/update | Actualizar pasajero no publicado asociado a un customer logeado como otro customer | Excepción unauthorise |
| customer/passenger/update | Actualizar pasajerosin estar logeado | Excepción unauthorise |
| customer/passenger/update | Actualizar pasajero logeado como administrador | Excepción unauthorise |
| customer/passenger/publish | Publicar pasajero valido asociado a un customer logeado como ese customer | Se publica el pasajero. |
| customer/passenger/publish | Publicar pasajero nulo | Excepción unauthorise |
| customer/passenger/publish | Publicar pasajero ya publicado | Excepción unauthorise |
| customer/passenger/publish | Publicar pasajero valido asociado a un customer logeado como otro customer | Excepción unauthorise |
| customer/passenger/publish | Publicar pasajero logeado como administrator | Excepción unauthorise |
| customer/passenger/publish | Publicar pasajero sin estar logeado | Excepción unauthorise |

# Performance testing (Ordenador 1)

Tras realizar las pruebas anteriores añadiendo índices se ha obtenido la siguiente tabla de tiempos medios agrupados por feature:





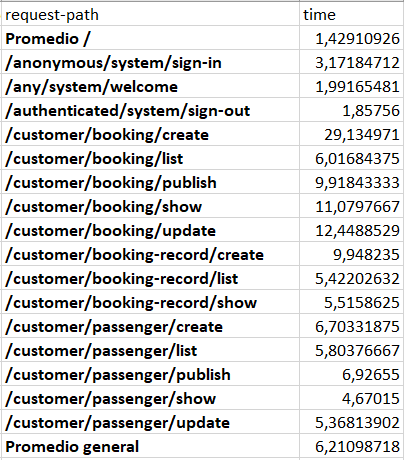
Se puede ver que la media ha bajado considerablemente. Además, la feature de /customer/booking/create que antes era la que mayor tiempo consumía ahora consume 10ms menos que antes. Ha bajado un 20% su consumo de tiempo.

|  |  |
| --- | --- |
| *Ordenador1* | |
|  |  |
| Media | 9,093341197 |
| Error típico | 0,4654495717 |
| Mediana | 5,0972 |
| Moda | 1,7846 |
| Desviación estándar | 11,25772325 |
| Varianza de la muestra | 126,7363327 |
| Curtosis | 9,321859834 |
| Coeficiente de asimetría | 2,714016729 |
| Rango | 84,6351 |
| Mínimo | 1,1452 |
| Máximo | 85,7803 |
| Suma | 5319,6046 |
| Cuenta | 585 |
| Nivel de confianza(95,0%) | 0,9141589616 |

|  |  |  |
| --- | --- | --- |
| After interval: |  |  |
| Interval(ms) | 8,179182235 | 10,00750016 |
| Interval(s) | 0,008179182235 | 0,01000750016 |

# Performance testing después de añadir índices (Ordenador 2)

Tras realizar las pruebas anteriores añadiendo índices se ha obtenido la siguiente tabla de tiempos medios agrupados por feature:



Se puede apreciar que el Ordenador2 tiene mayor capacidad computacional y baja aproximadamente unos 10ms cada feature a la cual se le implementó un índice.

|  |  |
| --- | --- |
| *Ordenador2* | |
|  |  |
| Media | 6,210987179 |
| Error típico | 0,3367875427 |
| Mediana | 4,3505 |
| Moda | 1,2104 |
| Desviación estándar | 8,145803928 |
| Varianza de la muestra | 66,35412163 |
| Curtosis | 13,28290207 |
| Coeficiente de asimetría | 3,398775188 |
| Rango | 61,7719 |
| Mínimo | 0,831 |
| Máximo | 62,6029 |
| Suma | 3633,4275 |
| Cuenta | 585 |
| Nivel de confianza(95,0%) | 0,6614623132 |

# Contraste de 95% de confianza comparación con otro ordenador

En esta sección se compara el rendimiento del Ordenador1 con el del Ordenador2. Ambos análisis se han hecho con los índices puestos.

|  |  |  |
| --- | --- | --- |
|  | *Ordenador1* | *Ordenador2* |
| Media | 9,093341197 | 6,210987179 |
| Varianza (conocida) | 126,736333 | 66,3541216 |
| Observaciones | 585 | 585 |
| Diferencia hipotética de las medias | 0 |  |
| z | 5,017008667 |  |
| P(Z<=z) una cola | 0,0000002624110051 |  |
| Valor crítico de z (una cola) | 1,644853627 |  |
| Valor crítico de z (dos colas)\* | 0,0000005248220103 |  |
| Valor crítico de z (dos colas) | 1,959963985 |  |

Nota: El valor Valor crítico de z (dos colas) que tiene puesto un “\*” suponemos que es equivalente al “P(Z<=z) two-tail” en inglés.

Definiendo un alfa = 0’05. Se observa que el valor de P(Z<=z) de dos colas es significativamente menor a alfa. Por tanto, podemos comparar las medias de los timpos de respuesta. Como el tiempo medio del Ordenador2 es menor que el del Ordenador1, el rendimiento es major en el Ordenador2

# Conclusión

En un proyecto es muy importante tener casos de pruebas definidos para confirmar que la aplicación desarrollada funciona como se espera y no dejar sin probar ningún caso. Para poder automatizar estas pruebas y no tener que realizarlas manualmente se hace uso de los tests, en nuestro caso test end-to-end.

No solo es importante testear que la aplicación funcione, sino que también hay que comprobar el rendimiento. Estas pruebas de rendimiento pueden ayudarnos a optimizar mucho nuestra aplicación y hacer que funcione sin operaciones ineficientes por tener métodos o consultas mal optimizadas. Todo esto repercute en la experiencia de usuario, a menor tiempo de respuesta mejor experiencia de usuario.

# Bibliografía

Diapositivas de Diseño y Pruebas 2 – Universidad de Sevilla