|  |
| --- |
| **Informe de Pruebas.**  **Entregable 4** |

Grupo: C1.02.14

Repositorio de GitHub: https://github.com/tcabgom/Acme-L3-D04

Fecha: 26/05/2023

Autor: Jesús Solís Ortega (Alumno #1)

Contenido

[Resumen 3](#_Toc135907760)

[Versión 3](#_Toc135907761)

[Introducción 3](#_Toc135907762)

[Pruebas Funcionales 3](#_Toc135907763)

[Pruebas de Rendimiento 4](#_Toc135907764)

[Conclusión 4](#_Toc135907765)

[Bibliografía 4](#_Toc135907766)

|  |
| --- |
| Resumen |

En este documento se va a documentar los requisitos que he implementado en este entregable, las decisiones y correcciones que he tomado a la hora de implementarlos.

|  |
| --- |
| Versión |

|  |  |  |
| --- | --- | --- |
| Número de revisión | Fecha | Descripción |
| 1.0 | 26/05/2023 | Creación del documento |

|  |
| --- |
| Introducción |

A continuación, se van a enumerar los requisitos implementados en este entregable, las modificaciones que se han realizado tras el análisis y un enlace a las validaciones.

|  |
| --- |
| Pruebas Funcionales |

El listado de muestra un listado de las pruebas realizadas es el siguiente:

* **PRUEBAS DE LISTADO:**
  + **LecturerCourseListTest:** Comprueba que la aplicación lista los cursos de los profesores de manera exitosa, y rechaza los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
  + **LecturerLectureListTest:** Comprueba que la aplicación lista correctamente las lecciones de un curso exitosamente, además de rechazar los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
  + **LecturerLectureListAllTest:** Comprueba que la aplicación lista correctamente las lecciones de un profesor exitosamente, además de rechazar los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
* **PRUEBAS DE MOSTRADO:**
  + **LecturerCourseShowTest:** Comprueba que la aplicación muestra correctamente la información de los cursos de los profesores, y rechaza los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
  + **LecturerLectureShowTest:** Comprueba que la aplicación muestra correctamente la información de las sesiones de las lecciones, y rechaza los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
* **PRUEBAS DE CREADO:**
  + **LecturerCourseCreateTest:** Comprueba que la aplicación crea correctamente nuevos cursos con parámetros correctos y no crea los tutoriales con parámetros incorrectos, además de rechazar los intentos de hackeo. Gracias a las pruebas, se localizó un error en la validación de los límites de los valores.
  + **LecturerLectureCreateTest:** Comprueba que la aplicación crea correctamente nuevas lecciones con parámetros correctos y no crea las lecciones con parámetros incorrectos, además de rechazar los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
  + **LecturerLecturesInCourseCreateTest:** Comprueba que la aplicación crea correctamente nuevas asociaciones entre un curso y una lección con parámetros correctos y no crea las asociaciones con parámetros incorrectos, además de rechazar los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
* **PRUEBAS DE ACTUALIZACIÓN:**
  + **LecturerCourseUpdateTest:** Comprueba que la aplicación actualiza correctamente cursos con parámetros correctos y no actualiza los cursos con parámetros incorrectos, además de rechazar los intentos de hackeo.
  + **LecturerLectureUpdateTest:** Comprueba que la aplicación actualiza correctamente lecciones con parámetros correctos y no actualiza las lecciones con parámetros incorrectos, además de rechazar los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
* **PRUEBAS DE BORRADO:**
  + **LecturerCourseDeleteTest:** Comprueba que la aplicación elimina los cursos de los profesores de manera exitosa, y rechaza los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
  + **LecturerLectureDeleteTest:** Comprueba que la aplicación elimina las lecciones de los profesores de manera exitosa, y rechaza los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
  + **LecturerLecturesInCourseDeleteTest:** Comprueba que la aplicación elimina las asociaciones entre cursos y lecciones de manera exitosa, y rechaza los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.
* **PRUEBA DE PUBLICADO:**
  + **LecturerCoursePublishTest:** Comprueba que la aplicación publica los cursos de los profesores de manera exitosa, y rechaza los intentos de hackeo. Gracias a las pruebas, se localizó un error que hacía que se pudiesen publicar cursos mediante hacking.
  + **LecturerLecturePublishTest:** Comprueba que la aplicación publica las lecciones de los profesores de manera exitosa, y rechaza los intentos de hackeo. No se encontraron errores durante la fase de testeo al ejecutarlos.

|  |  |
| --- | --- |
| Pruebas de Rendimiento |  |

A continuación, se muestran dos gráficas que indican el tiempo promedio tomado por cada prueba en ser ejecutada en dos ordenadores distintos. **A picture containing text, screenshot, font, line

Description automatically generated**

**A picture containing text, screenshot, font, line

Description automatically generated**

Como podemos observar, el tiempo tomado en ejecutar las pruebas es casi el mismo en ambos ordenadores, con una diferencia bastante notable en course/delete y lectures-in-course/create donde la diferencia es más del doble de tiempo en el ordenador 2. Esto último resulta especialmente llamativo dado el hecho de que el ordenador 2 es significativamente más potente que el ordenador 1, lo cual puede deberse a problemas de compatibilidad con el hardware.

La siguiente gráfica muestra la media de los tiempos de ejecución de las pruebas agrupadas por “request path”.

A picture containing text, line, screenshot, plot

Description automatically generated

A screenshot of a graph

Description automatically generated with medium confidence

A picture containing text, screenshot, line, plot

Description automatically generated

Por otro lado, se notan picos notables en el tiempo medio que tardan los test de hacking y en mucho menor grado, las pruebas positivas. En términos generales los tiempos no son demasiado dispares entre pruebas del mismo tipo salvo algunas pruebas aisladas (“outliers”) que coincide de nuevo en un hacking, lo cual está dentro de la generalidad que presentan los datos.

Para esta última prueba de rendimiento se han llevado a cabo una serie de cambios para probar si mejora el rendimiento.

|  |  |  |
| --- | --- | --- |
| z-Test: Two Sample for Means |  |  |
|  |  |  |
|  | *Versión 2 (actualizada)* | *Versión 1 (previa)* |
| Mean | 4,458840372 | 5,678756477 |
| Known Variance | 9,545045359 | 66,63302245 |
| Observations | 1397 | 1351 |
| Hypothesized Mean Difference | 0 |  |
| z | -5,148022361 |  |
| P(Z<=z) one-tail | 1,31624E-07 |  |
| z Critical one-tail | 1,644853627 |  |
| P(Z<=z) two-tail | 2,63247E-07 |  |
| z Critical two-tail | 1,959963985 |  |

Se puede ver que los resultados del Z-Test, nos da unos valores dentro de los aceptable al comparar los resultados, ya que los P(Z<=z) está dentro del intervalo de confianza y son por tanto comparables. Por tanto, el rendimiento es mejor en la versión 2 gracias a la actualización.

|  |  |
| --- | --- |
| Conclusión |  |

En este documento se han presentado los requisitos implementados en el entregable, así como las pruebas funcionales realizadas y los resultados obtenidos. También se han incluido pruebas de rendimiento y un análisis comparativo entre dos ordenadores.

En cuanto a las pruebas funcionales, se han realizado pruebas de listado, mostrado, creado, actualización, borrado y publicado, tanto para cursos como para lecciones. En general, no se encontraron errores durante la fase de testeo y las pruebas fueron exitosas, demostrando que la aplicación cumple con los requisitos establecidos.

En relación con las pruebas de rendimiento, se han presentado gráficas que muestran el tiempo promedio tomado por cada prueba en dos ordenadores distintos. Se observa que el tiempo de ejecución de las pruebas es similar en ambos ordenadores, aunque con algunas diferencias notables en ciertos casos. Además, se ha realizado un z-Test de dos muestras para comparar los tiempos promedio tras actualizar el programa para mejorar el rendimiento, y se ha determinado que los resultados son comparables y el rendimiento es mejor en la versión actualizada.

En conclusión, este documento proporciona una visión general de los requisitos implementados, las pruebas funcionales realizadas y los resultados obtenidos. Además, se ha realizado un análisis de rendimiento que muestra mejoras en el tiempo de ejecución. Esto demuestra el éxito en la implementación de los requisitos y la calidad de la aplicación desarrollada.

|  |
| --- |
| Bibliografía |

*Intencionalmente en blanco.*