Student #4, Sprint 4: Testing Report

Group: C1.04.14

Repository: https://github.com/marizqlav/Acme-L3-D04

Student #1		**Student #2**	
Name:	Domínguez-Adame, Alberto	**Name:**	Herrera Ramírez, Ismael
email:	albdomrui@alum.us.es	**email:**	ismherram@alum.us.es
Student #3		**Student #4**	
Name:	Olmedo Marín, Marcos	**Name:**	Izquierdo Lavado, Mario
email:	marolmmar1@alum.us.es	**email:**	marizqlav @alum.us.es
Student #5			
Name:	Merino Palma, Alejandro		
email:	alemerpal@alum.us.es		

Table of contents

Summary

Acme Life-Long Learning, Inc. (Acme L3, Inc. for short) is a company that specializes in helping students get started in a variety of subjects with the help of renowned instructors. The goal of this project is to develop a WIS to help this organization run its business.

Next, I analyzed the performance of the tests on my computer for the documents generated automatically when running the tests: request-performance and test-performance.

Using the Excel data analysis tool, we were able to compare the performance obtained by each team. To do this, we performed two analyses: descriptive statistics and the z test.

For a better overview of the analysis, we generated several graphs with the average access times.

Revision table

Number	Date	Description
1	21/05/2023	Full redaction of the document

Introduction

This document lists the individual tasks assigned to the student during the fourth spring of development on Acme L3.

This document presents the analysis and comparison of the performance of the project's tests on two different computers. We have used a statistical analysis and a hypothesis test.

I have divided the document according to the test carried out and the computer involved. Therefore, the structure that consists would be distributed as follows:

- Contents
  - Functional testing
  - Performance Tests
    - Evolutionary graphs
      - Computer feature
    - PC - Performance Request
    - PC - Performance Test Case
    - Statistic analysis
- Conclusion
- Bibliography

Once I reach the end of the document, I collect the result of the analysis carried out in the conclusion.

Contents

Functional testing

Due to the individual nature of this report, the team member designated to each task will be omitted. During the Sprint 4, Mario Izquierdo has implemented the following test cases, grouped by functionality. For each test case, a succinct description will be provided plus a clear indication of how effective it was in detecting errors.

Practicum Test

- **Create:**
    - **positive:**

Create a practicum and let you enter the empty list of sessions of that practicum

    - **negative:**

The practicum is not created because the restrictions are not met

    - **hacking:**

This test tries to create a practicum using principals with inappropriate roles

- **Delete:**
    - **positive:**

Practicums are successfully eliminated

    - **negative:**

It is not removed because those practicums are published and a published practicum cannot be deleted

    - **hacking:**

This test tries to delete a company with a role other than "Company".

- **ListAll:**
    - **positive:**

This test signs in as a company, lists all of the practicums, and then checks that the listing shows the expected data.

    - **negative:**

There aren't any negative tests for this feature because it's a listing

    - **hacking:**

This test tries to list all of the practicums using inappropriate roles.

- **Publish:**
    - **positive:**

    The practicums are published in a correct way

    - **negative:**

    This test attempts to publish a practicum that cannot be published, yet.

    - **hacking300:**

    This test tries to publish a practicum with a role other than "Employer".

    - **hacking301:**

    This test tries to publish a published practicum that was registered by the principal.

    - **hacking302:**

    This test tries to publish a practicum that wasn't registered by the principal, be it published or unpublished.

- **Show:**
    - **positive:**

    This test signs in as a company, lists all of the practicums, clicks on one of them, and checks that the form has the expected data.

    - **negative:**

    There aren't any negative tests for this feature because it's a listing that doesn't involve entering any data in any forms.

    - **hacking:**

    This test tries to show an unpublished practicum by someone who is not the principal. And, being logged in as companyX not being able to see the details of a recipe that is not yours.

- **Update:**
    - **positive:**

    Practicum is well updated

    - **negative:**

    This test tries to update a practicum because the constraints are not met

    - **hacking:**

    This test tries to show an unupdated a practicum by someone who is not the principal. And, being logged in as companyX not being able to see the details of a practicum that is not yours.

Session Practicum Test

- **Create Addendum:**
    - **positive:**

      Create an addendum session practicum
    - **negative:**

      The addendum session practicum is not created because the practicum is not published
    - **negative2:**

      The addendum session practicum is not created because the restrictions are not met
    - **hacking300:**

      This test tries to create a addendum session practicum for a practicum as a principal without the "Company" role
    - **hacking301:**

      This test tries to create an addendum session practicum for a published practicum created by the principal.
    - **hacking302:**

      This test tries to create an addendum session practicums for practicums that weren't created by the principal.

- **Create:**
    - **positive:**

      Create a session practicum
    - **negative:**

      The session practicum is not created because the restrictions are not met
    - **hacking300:**

      This test tries to create a session practicum for a practicum as a principal without the "Company" role
    - **hacking301:**

      This test tries to create a session practicum for a published practicum created by the principal.
    - **hacking302:**

      This test tries to create session practicums for practicums that weren't created by the principal.

- **Delete:**
    - **positive:**

      Session practicums are successfully eliminated
    - **negative:**

      It is not removed because those session practicums are published and a published session practicum cannot be deleted
    - **hacking:**

      This test tries to delete a company with a role other than "Company".

- **ListAll:**
    - **positive:**

      This test signs in as a company, lists all of the practicums, clicks on one of them and then checks that the listing shows the expected data.
    - **negative:**

      There aren't any negative tests for this feature because it's a listing
    - **hacking:**

      This test tries to list all of the session practicums using inappropriate roles.

- **Publish:**
    - **positive:**

      The session practicums are published in a correct way
    - **negative:**

      This test attempts to publish a session practicum that cannot be published, yet.
    - **hacking:**

      This test tries to publish a session practicum with a role other than "Company".

- **Show:**
    - **positive:**

      This test signs in as a company, lists all of the practicums, clicks on one of them, next, clicks on one of the session practicum and checks that the form has the expected data.
    - **negative:**

      There aren't any negative tests for this feature because it's a listing that doesn't involve entering any data in any forms.
    - **hacking:**

      This test tries to show an unpublished session practicum by someone who is not the principal. And, being logged in as companyX not being able to see the details of a session practicum that is not yours.

- **Update:**
    - **positive:**

    Session Practicum of a practicum is well updated.

    - **negative:**

    This test tries to update session practicums because the constraints are not met.

    - **hacking:**

    This test tries to show an unupdated session practicum by someone who is not the principal. And, being logged in as companyX not being able to see the details of a session practicum that is not yours.

Performance Tests

Chapter on performance testing: You should provide proper graphs and a 95% confidence interval for the wall time your project takes to serve requests in your functional tests on a two-time computer, before doing the refactoring and after. the refactoring, plus a 95% confidence hypothesis test on what is the best state for the code. This is the scheme that will be followed for both the practicum and the practicum session:

- Evolutionary graphs
    - Computer feature
- PC - Performance Request
- PC - Performance Test Case
    - Statistic analysis

Once we reach the end of the document, we collect the result of the analysis carried out in the conclusion.

Evolutionary graphs(Practicum)

A comparison of the average response time of the GET and POST requests made to the server between two different machines has been carried out. It can be seen that the graphs vary depending on the characteristics of said equipment.

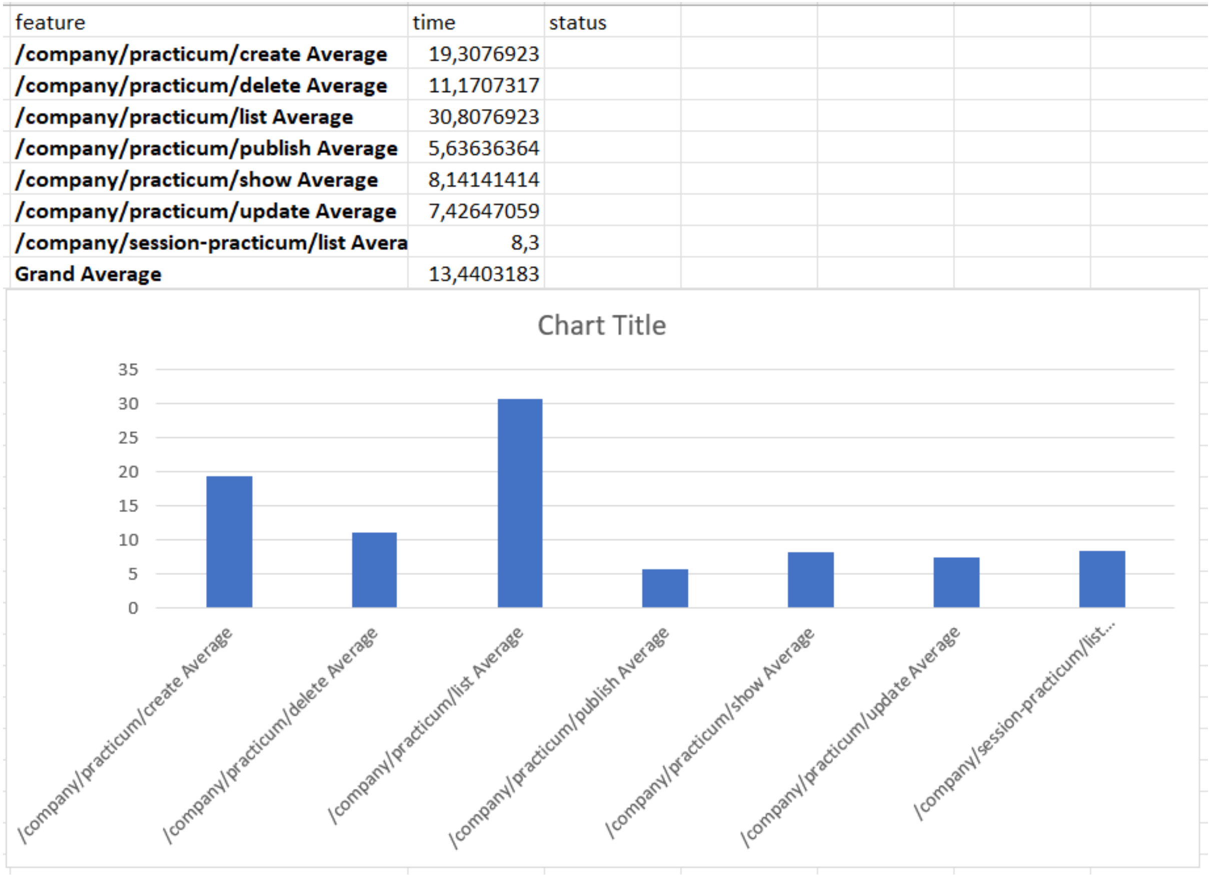
Computer Feature:

Nombre de dispositivo: DESKTOP-76JUKRD
Procesador: 12th Gen Intel(R) Core(TM) i7-12700H 2.70 GHz
RAM instalada: 32,0 GB (31,7 GB usable)

Below are the graphs of the request file executed on the machine

PC – Performance Request

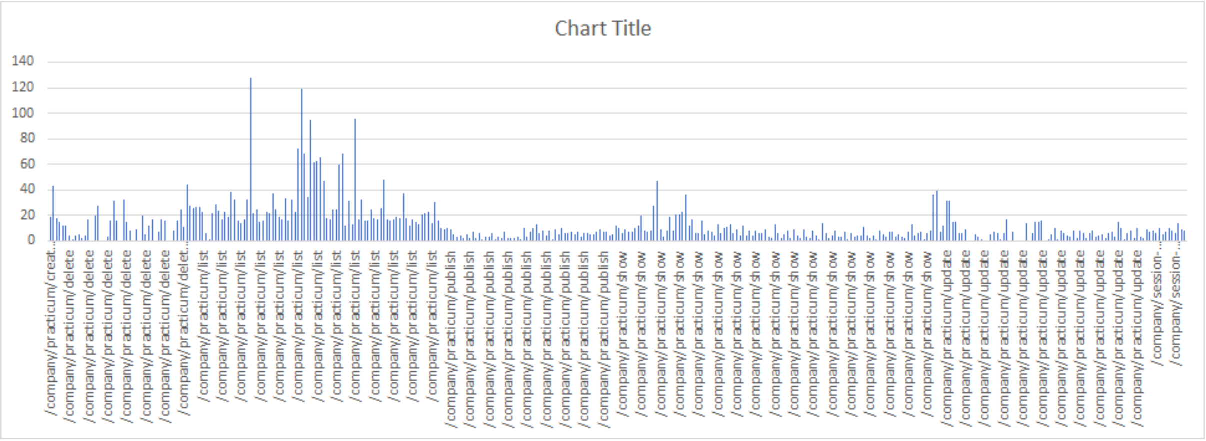
The following chart intuitively clarifies which functions are the most time consuming:



feature	time	status
/company/practicum/create Average	19,3076923	
/company/practicum/delete Average	11,1707317	
/company/practicum/list Average	30,8076923	
/company/practicum/publish Average	5,63636364	
/company/practicum/show Average	8,14141414	
/company/practicum/update Average	7,42647059	
/company/session-practicum/list Avera	8,3	
Grand Average	13,4403183	

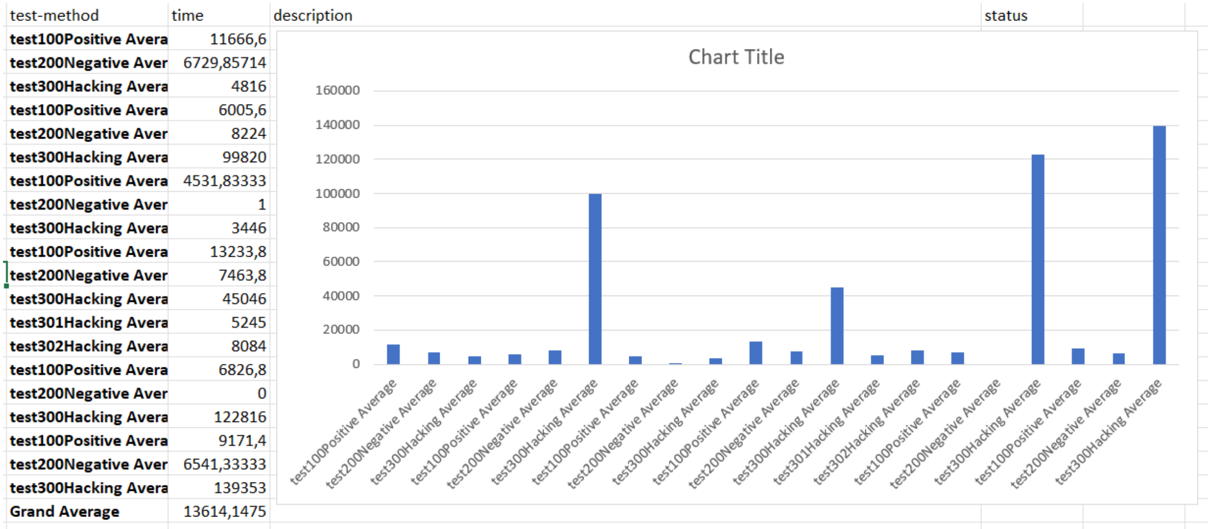
Chart Title
35
30
25
20
15
10
5
0
/company/practicum/create Average
/company/practicum/delete Average
/company/practicum/list Average
/company/practicum/publish Average
/company/practicum/show Average
/company/practicum/update Average
/company/session-practicum/list...

We have obtained the following graph of displaying each of the tests:

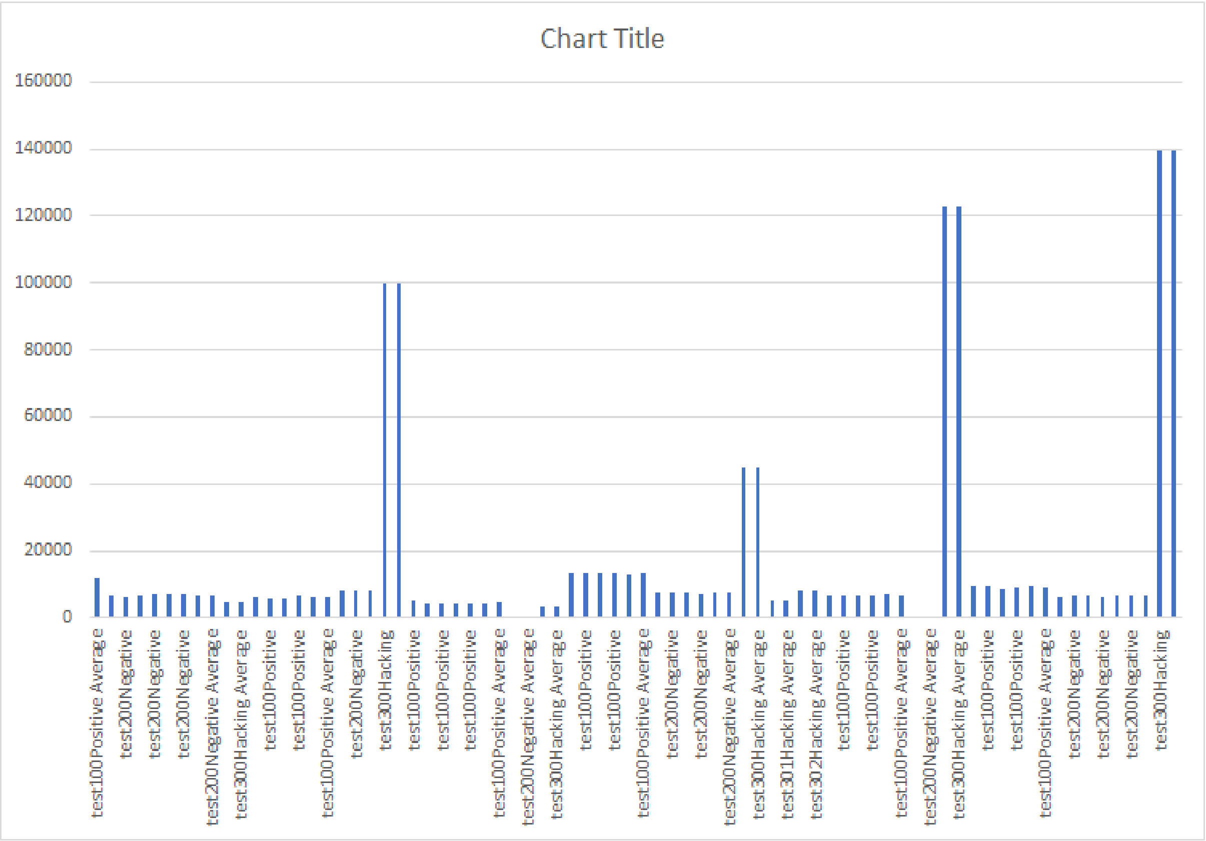


Intuitively, the conclusion is that most test cases take anywhere from about zero seconds to 130 seconds to complete.

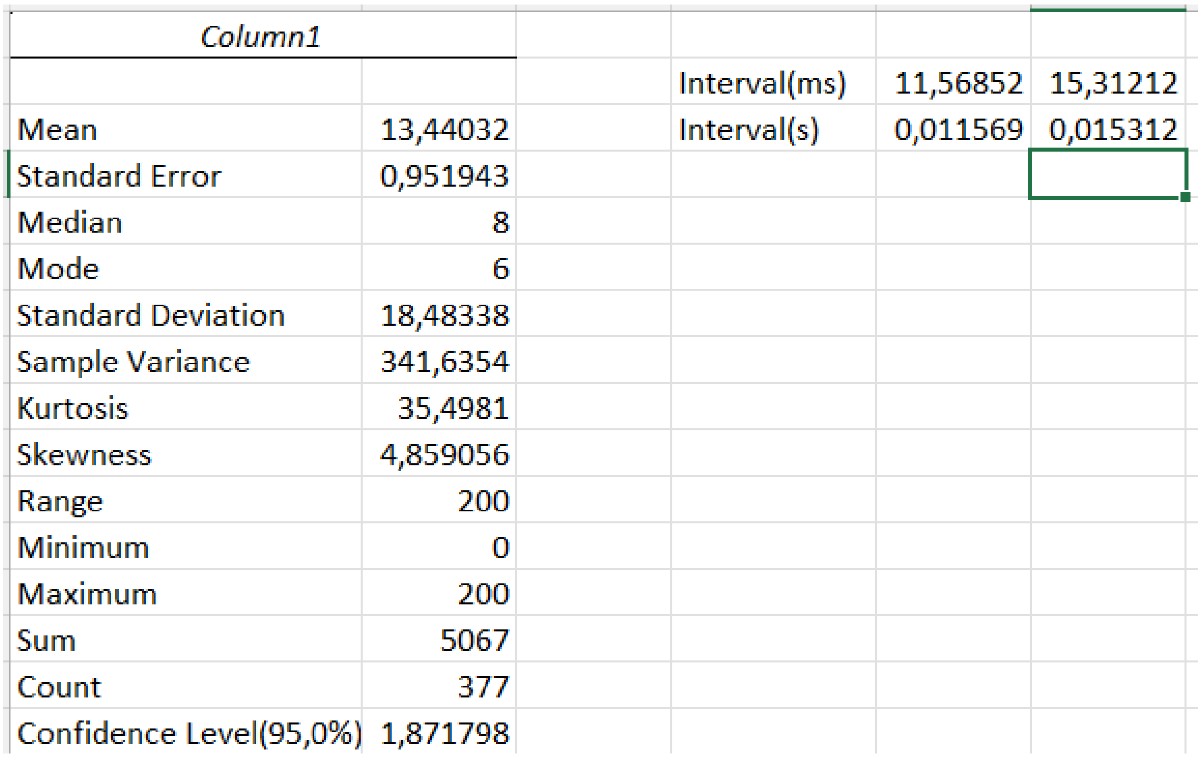
Finally, graph your log. Intuitively, the conclusion is that most test cases take from roughly zero seconds up to 140 seconds to complete.



test-method	time	description	status
test100Positive Avera	11666,6		
test200Negative Aver	6729,85714		
test300Hacking Avera	4816		
test100Positive Avera	6005,6		
test200Negative Aver	8224		
test300Hacking Avera	99820		
test100Positive Avera	4531,83333		
test200Negative Aver	1		
test300Hacking Avera	3446		
test100Positive Avera	13233,8		
test200Negative Aver	7463,8		
test300Hacking Avera	45046		
test301Hacking Avera	5245		
test302Hacking Avera	8084		
test100Positive Avera	6826,8		
test200Negative Aver	0		
test300Hacking Avera	122816		
test100Positive Avera	9171,4		
test200Negative Aver	6541,33333		
test300Hacking Avera	139353		
Grand Average	13614,1475		



In addition, an auxiliary scheme has been used that shows the different intervals in a very visual way:



Column1					
			Interval(ms)	11,56852	15,31212
Mean	13,44032		Interval(s)	0,011569	0,015312
Standard Error	0,951943				
Median	8				
Mode	6				
Standard Deviation	18,48338				
Sample Variance	341,6354				
Kurtosis	35,4981				
Skewness	4,859056				
Range	200				
Minimum	0				
Maximum	200				
Sum	5067				
Count	377				
Confidence Level(95,0%)	1,871798				

PC – Performance Test Case

Due to lack of time, the refactoring necessary to perform the comparisons could not be carried out.

Evolutionary graphs(Sesion Practicum)

A comparison of the average response time of the GET and POST requests made to the server between two different machines has been carried out. It can be seen that the graphs vary depending on the characteristics of said equipment.

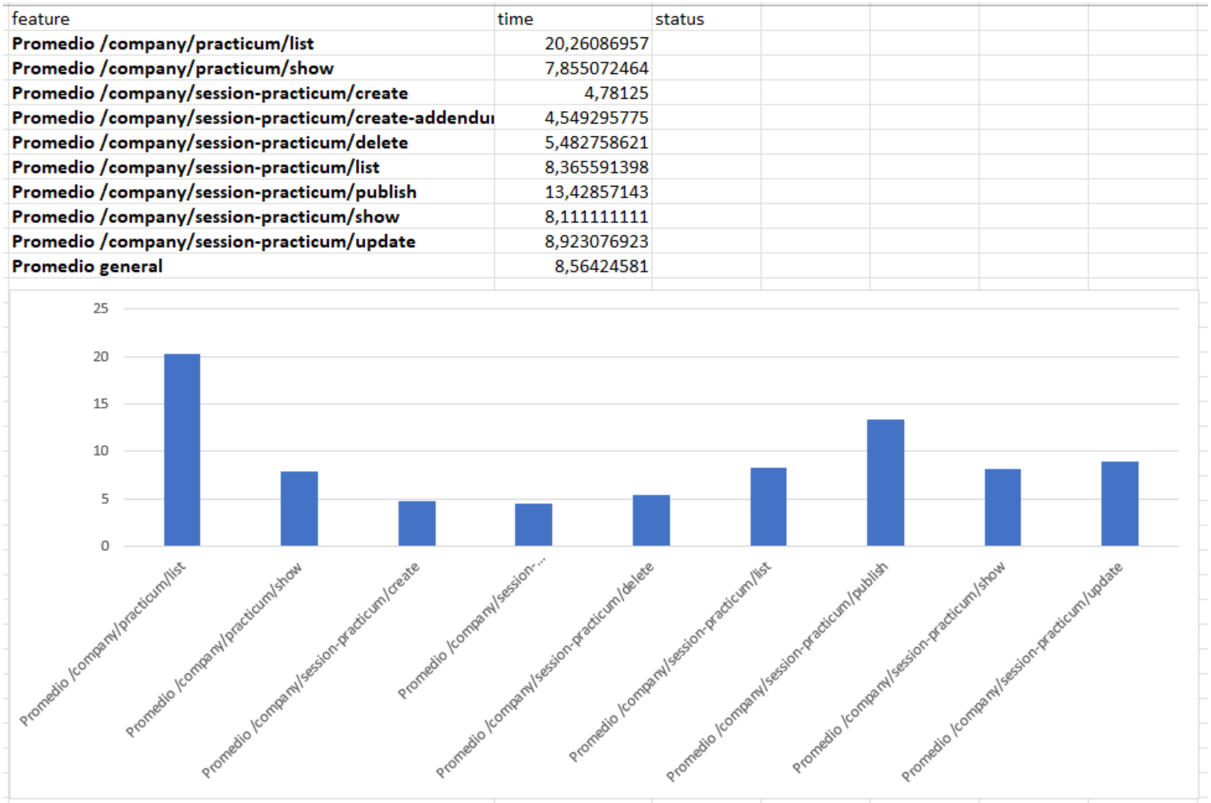
Computer Feature:

Nombre de dispositivo: DESKTOP-76JUKRD
Procesador: 12th Gen Intel(R) Core(TM) i7-12700H 2.70 GHz
RAM instalada: 32,0 GB (31,7 GB usable)

Below are the graphs of the request file executed on the machine

PC – Performance Request

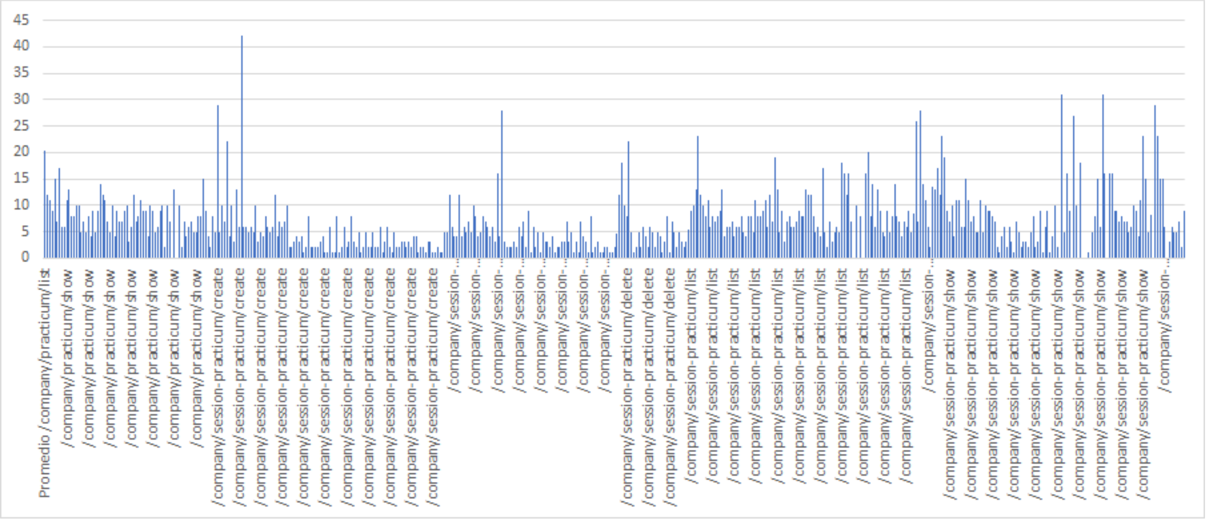
The following chart intuitively clarifies which functions are the most time consuming:



feature	time	status				
Promedio /company/practicum/list	20,26086957					
Promedio /company/practicum/show	7,855072464					
Promedio /company/session-practicum/create	4,78125					
Promedio /company/session-practicum/create-addendu	4,549295775					
Promedio /company/session-practicum/delete	5,482758621					
Promedio /company/session-practicum/list	8,365591398					
Promedio /company/session-practicum/publish	13,42857143					
Promedio /company/session-practicum/show	8,111111111					
Promedio /company/session-practicum/update	8,923076923					
Promedio general	8,56424581					

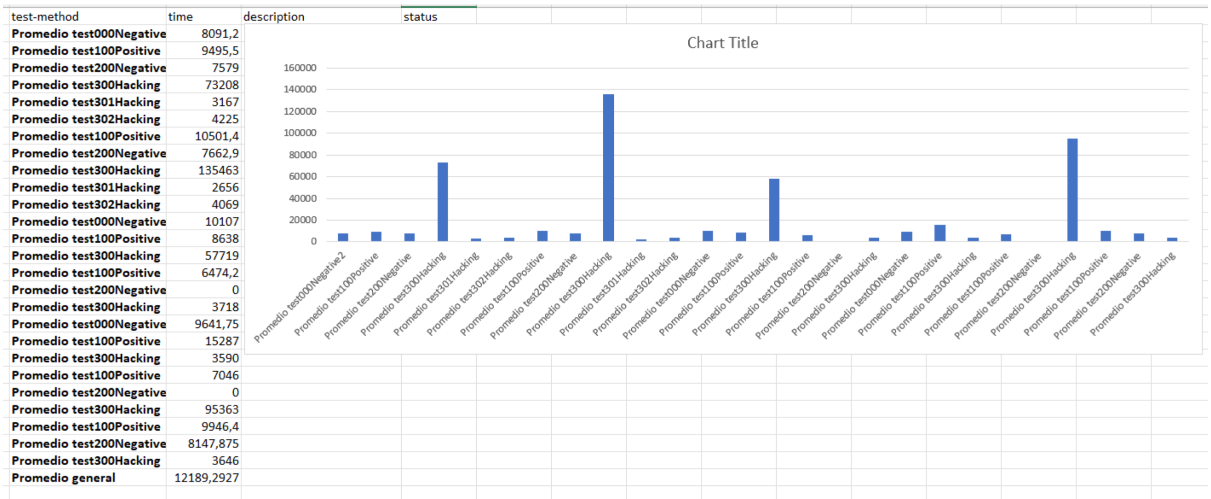
25
20
15
10
5
0
Promedio /company/practicum/list
Promedio /company/practicum/show
Promedio /company/session-practicum/create
Promedio /company/session-…
Promedio /company/session-practicum/delete
Promedio /company/session-practicum/list
Promedio /company/session-practicum/publish
Promedio /company/session-practicum/show
Promedio /company/session-practicum/update

We have obtained the following graph of displaying each of the tests:

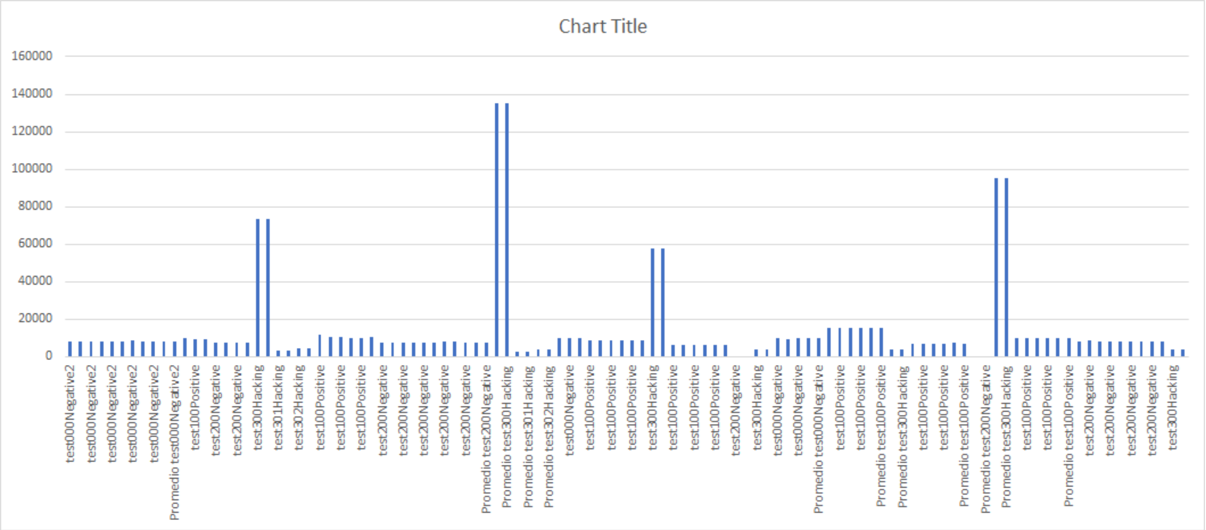


Intuitively, the conclusion is that most test cases take anywhere from about zero seconds to 45 seconds to complete.

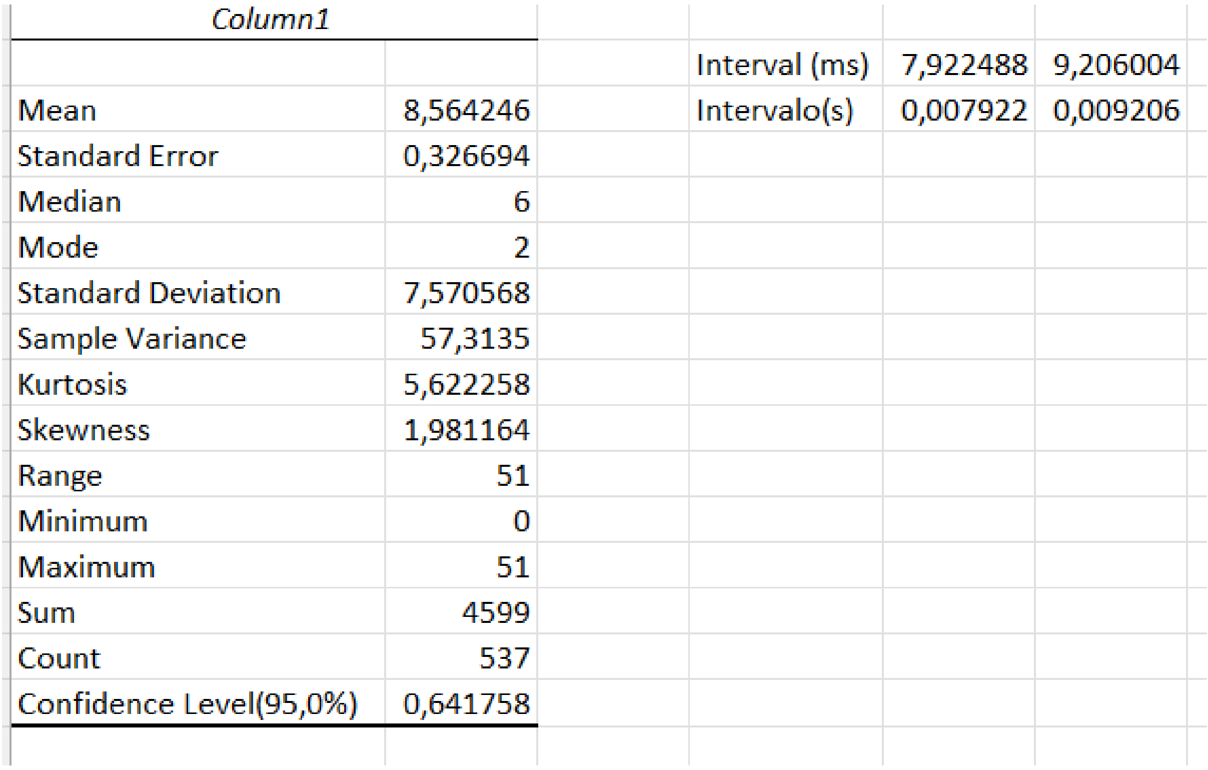
Finally, graph your log. Intuitively, the conclusion is that most test cases take from roughly zero seconds up to 135 seconds to complete.



test-method	time	description	status
Promedio test000Negative	8091,2		
Promedio test100Positive	9495,5		
Promedio test200Negative	7579		
Promedio test300Hacking	73208		
Promedio test301Hacking	3167		
Promedio test302Hacking	4225		
Promedio test100Positive	10501,4		
Promedio test200Negative	7662,9		
Promedio test300Hacking	135463		
Promedio test301Hacking	2656		
Promedio test302Hacking	4069		
Promedio test000Negative	10107		
Promedio test100Positive	8638		
Promedio test300Hacking	57719		
Promedio test100Positive	6474,2		
Promedio test200Negative	0		
Promedio test300Hacking	3718		
Promedio test000Negative	9641,75		
Promedio test100Positive	15287		
Promedio test300Hacking	3590		
Promedio test100Positive	7046		
Promedio test200Negative	0		
Promedio test300Hacking	95363		
Promedio test100Positive	9946,4		
Promedio test200Negative	8147,875		
Promedio test300Hacking	3646		
Promedio general	12189,2927		



In addition, an auxiliary scheme has been used that shows the different intervals in a very visual way:



Column1					
			Interval (ms)	7,922488	9,206004
Mean	8,564246		Intervalo(s)	0,007922	0,009206
Standard Error	0,326694				
Median	6				
Mode	2				
Standard Deviation	7,570568				
Sample Variance	57,3135				
Kurtosis	5,622258				
Skewness	1,981164				
Range	51				
Minimum	0				
Maximum	51				
Sum	4599				
Count	537				
Confidence Level(95,0%)	0,641758				

PC – Performance Test Case

Due to lack of time, the refactoring necessary to perform the comparisons could not be carried out.

Conclusion

In conclusion, I think that the tests can improve with the refactoring, as can be seen in the group, but due to the lack of time and that they wanted to deliver it by May, there was no time to carry out the actions.

Bibliography

- Document “On Your Derivables” of EV of the subject Design and Testing II.
- Document “So6 - On your follow-ups” of EV of the subject Design and Testing II.
- Document “o8- Annexes” of EV of the subject Design and Testing II.