Student #2, Sprint 4: Testing Report

Group: C1.04.14

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

Student #1

Name: Domínguez-Adame, Alberto
email: albdomrui@alum.us.es

Student #2

Name: Herrera Ramírez, Ismael
email: ismherram@alum.us.es

Student #3

Name: Olmedo Marín, Marcos
email: marolmmar1@alum.us.es

Student #4

Name: Izquierdo Lavado, Mario
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	26/05/2023	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, Ismael Herrera Ramirez 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.

Enrolment Test

- **Create:**
    - **Positive:** This test authenticates as a student and then lists his or her enrolments, creates a new one, and checks that it's been created properly.

    - **Positive 2:** This test authenticates you as a student and then lists the available courses, creates an enrollment from the course you want, and verifies that it was created successfully.

    - **Negative:** This test attempts to create enrolments with incorrect data.

    - **Hacking:** This test tries to create an enrolment using principals with inappropriate roles.

- **Delete:**
    - **Positive:** This test login as a principal with "Student" role and try to delete the indicated enrolment.

    - **Hacking:** This test login as a principal with "Student" role and try to delete the indicated enrolment.

- **ListMine:**
    - **Positive:** This test authenticates as a student, lists his or her enrolments only, and then checks that the listing has the expected data.

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

    - **Hacking:**This test tries to list enrolments using inappropriate roles.

- **Finalise:**
    - **Positive:** This test authenticates as a student, lists his or her enrolments, then selects one of them, and finalise it.

    - **Negative:**This test attempts to finalise an enrolment that cannot be finalised, yet.

    - **Hacking300:** This test tries to finalise an enrolment with a role other than "Student" or using a student who is not the owner.

    - **Hacking301:** This test tries to finalise a finalised enrolment that was registered by the principal.

    - **Hacking302:** This test tries to finalise an enrolment that wasn't registered by the principal, be it finalised or not finalised.

- **Show:**
    - **Positive:** This test signs in as a student, lists all of the enrolments, click 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 enrolment by someone who is not the principal.

- **Update:**
    - **Positive:** This test logs in as a student, lists his or her enrolments, selects one of them, updates it, and then checks that the update has actually been performed.

    - **Negative:** This test attempts to update an enrolment with wrong data.

    - **Hacking:** This test tries to update an enrolment with a role other than "Student", or using a student who is not the owner.

Activity Test

- **Create:**
    - **Positive:** This test authenticates as an student, list his or her jobs, navigates to their activities, and checks that they have the expected data.

    - **Negative:** This test attempts to create activities using wrong data.

    - **Hacking300:** This test tries to create an activity for a enrolment as a principal without the "Student" role.

    - **Hacking301:** This test tries to create an activity for a not finalised enrolment created by the principal.

    - **Hacking302:** This test tries to create activities for enrolments that weren't created by the principal.

- **Delete:**
    - **Positive:** This test login as a principal with "Student" role and try to delete the indicated activity

    - **Hacking:** This test tries to delete an activity with a role other than "Student",or using a student who is not the owner.

- **ListMine:**
    - **Positive:** This test authenticates as a student, selects an enrolment and lists the activities of these enrolment only, and then checks that the listing has the expected data.

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

    - **Hacking:** This test tries to list the activities of an enrolment that is not finalised using a principal that didn't create it.

- **Show:**
    - **Positive:** This test signs in as a student, lists his or her activities, selects one of them and checks that it's as expected.

    - **Negative:** There's no negative test case for this listing, since it doesn't involve filling in any forms.

    - **Hacking:** This test tries to show an activity using a principal that didn't create it.

- **Update:**
    - **Positive:** This test logs in as a student, lists his or her activities, selects one of them, updates it, and then checks that the update has actually been performed.

    - **Negative:** This test attempts to update an activity with wrong data.

    - **Hacking:** This test tries to update an activity with a role other than "Student", or using a student who is not the owner.

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 “enrolment” and “activity”:

- 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(Enrolment)

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:

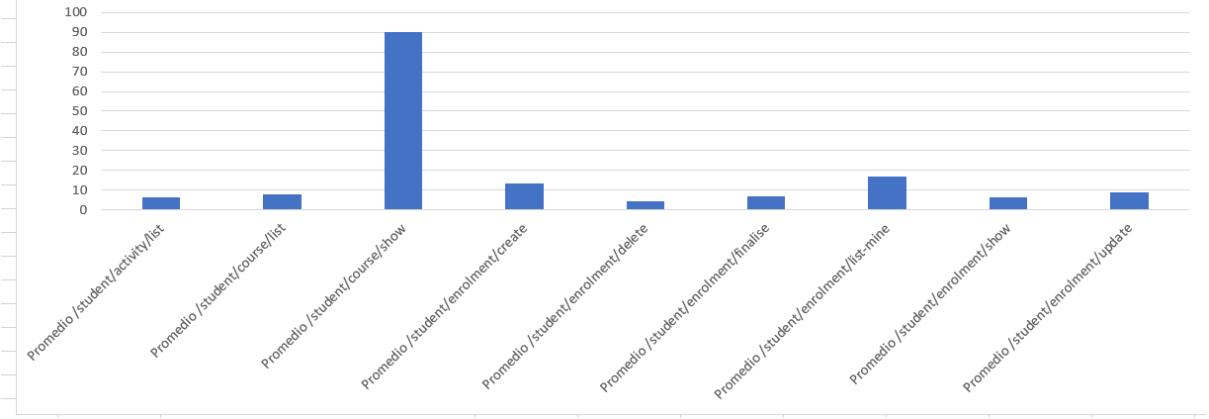
Nombre de dispositivo: DESKTOP-DICBMLN
Procesador: AMD Ryzen 7 3750H with Radeon
RAM instalada: 16,0 GB (15,4 GB usable)

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

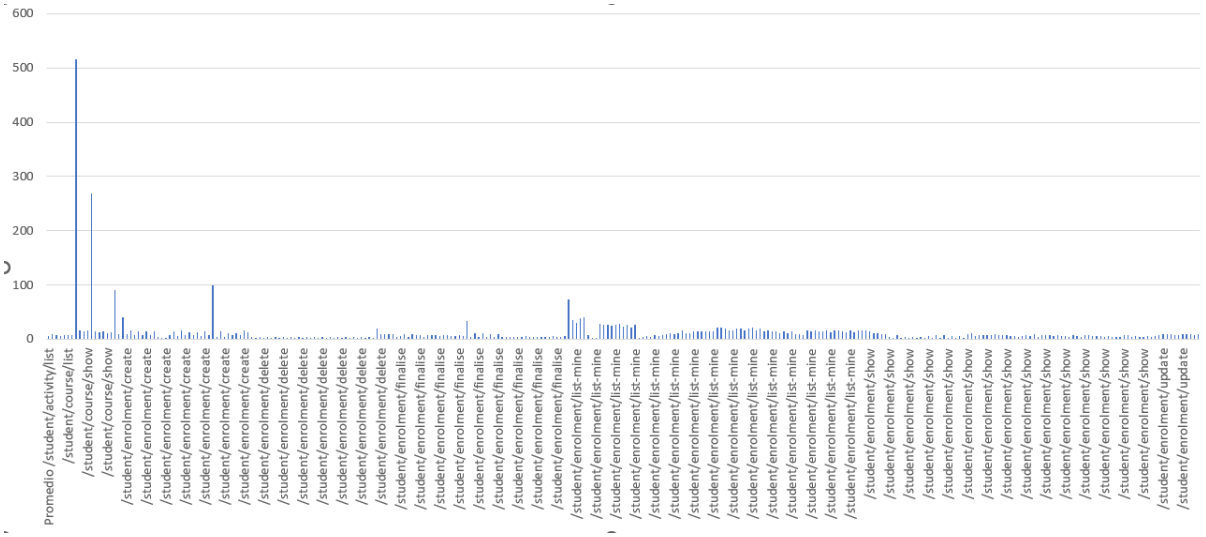
PC – Performance Request

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

Promedio /student/activity/list	6,2
Promedio /student/course/list	7,8
Promedio /student/course/show	89,9
Promedio /student/enrolment/create	13,4242424
Promedio /student/enrolment/delete	4,32432432
Promedio /student/enrolment/finalise	6,83333333
Promedio /student/enrolment/list-mine	17,0526316
Promedio /student/enrolment/show	6,15068493
Promedio /student/enrolment/update	8,9
Promedio general	12,6907216



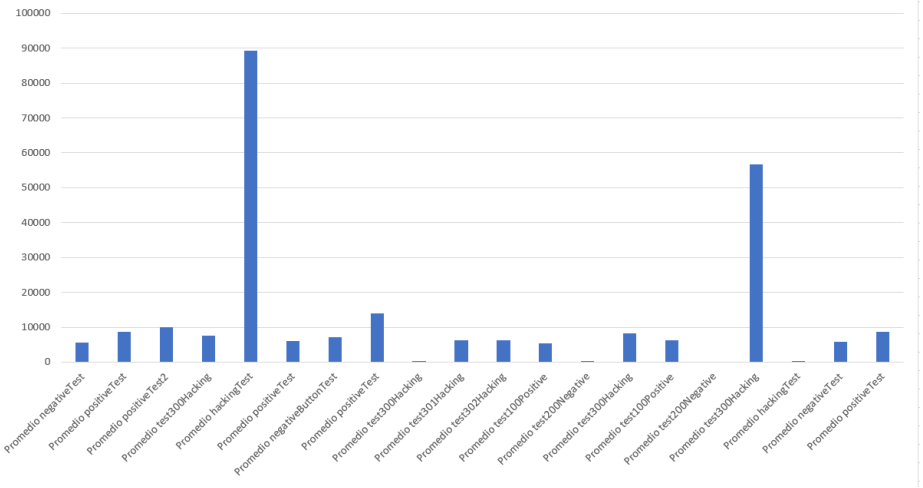
We have obtained the following graph of displaying each of the tests:

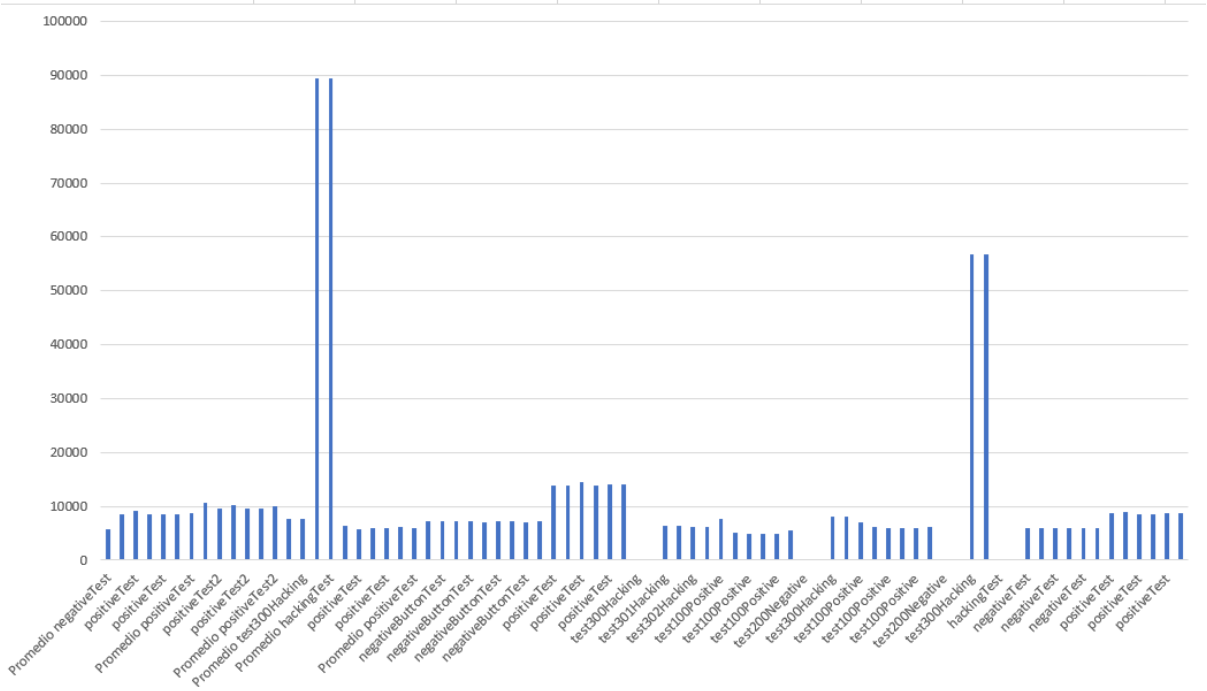


Intuitively, the conclusion is that most test cases take anywhere from about zero seconds to 510 seconds to complete.

Finally, graph your log. Intuitively, the conclusion is that most test cases take from roughly zero seconds up to 90 seconds to complete.

Promedio negativeTest	5702,8
Promedio positiveTest	8668,2
Promedio positiveTest2	9967,2
Promedio test300Hacking	7666
Promedio hackingTest	89353
Promedio positiveTest	6067,2
Promedio negativeButtonTest	7170,875
Promedio positiveTest	14040,2
Promedio test300Hacking	11
Promedio test301Hacking	6310
Promedio test302Hacking	6186
Promedio test100Positive	5475,8
Promedio test200Negative	1
Promedio test300Hacking	8204
Promedio test100Positive	6179,2
Promedio test200Negative	0
Promedio test300Hacking	56675
Promedio hackingTest	1
Promedio negativeTest	5927,2
Promedio positiveTest	8704
Promedio general	9292,587302





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

Columna1					
Media	12,6907216		Interval(ms)	8,70442195	16,6770213
Error típico	2,02537571		Interval(s)	0,00870442	0,01667702
Mediana	7				
Moda	5				
Desviación es	34,5503214				
Varianza de la	1193,72471				
Curtosis	166,177979				
Coeficiente de	12,2106555				
Rango	515				
Mínimo	1				
Máximo	516				
Suma	3693				
Cuenta	291				
Nivel de confi	3,9862997				

PC – Performance Test Case

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

Evolutionary graphs(Activity)

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:

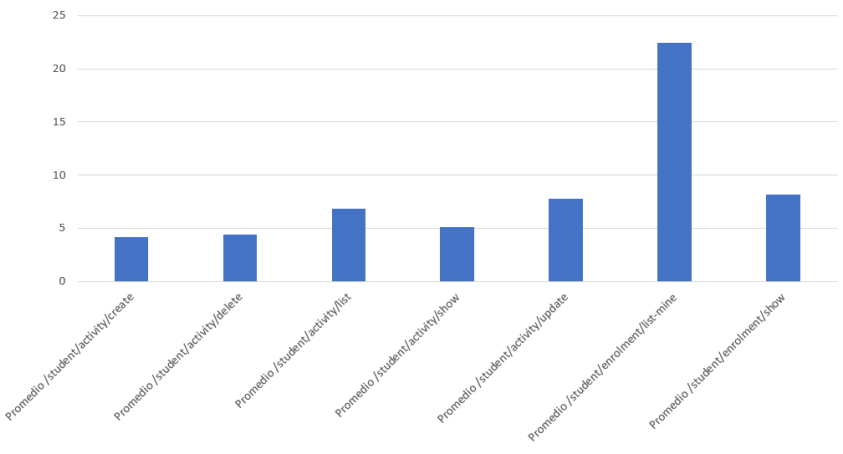
Nombre de dispositivo: DESKTOP-DICBMLN
Procesador: AMD Ryzen 7 3750H with Radeon
RAM instalada: 16,0 GB (15,4 GB usable)

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

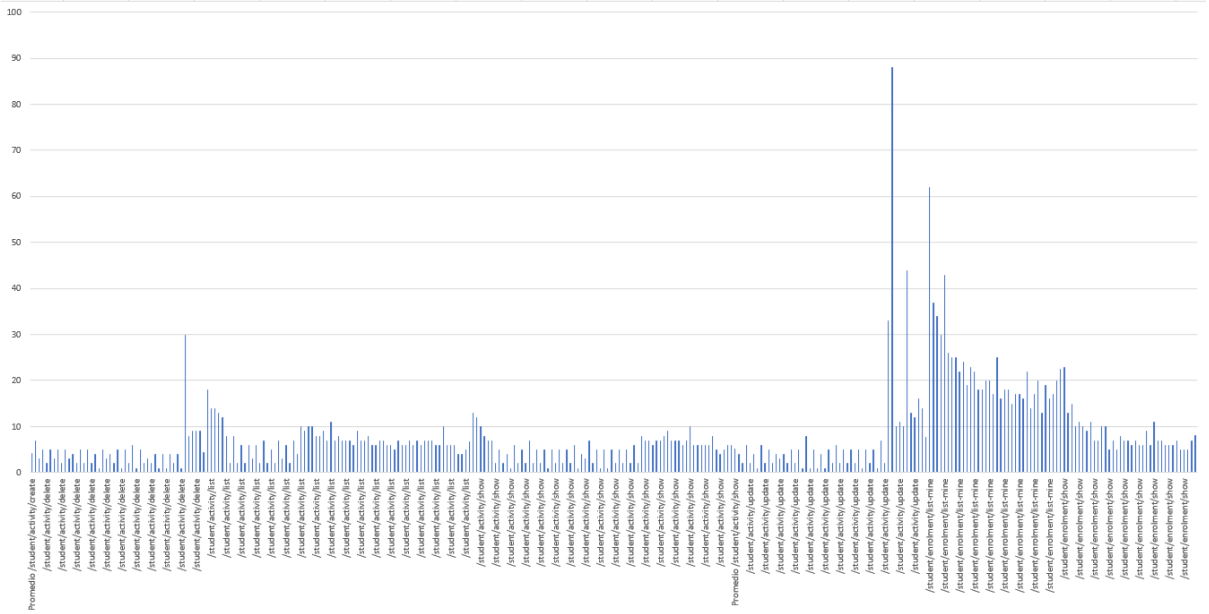
PC – Performance Request

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

feature	time
Promedio /student/activity/create	4,16666667
Promedio /student/activity/delete	4,37777778
Promedio /student/activity/list	6,82857143
Promedio /student/activity/show	5,14285714
Promedio /student/activity/update	7,76
Promedio /student/enrolment/list-mine	22,4285714
Promedio /student/enrolment/show	8,2
Promedio general	7,4137931



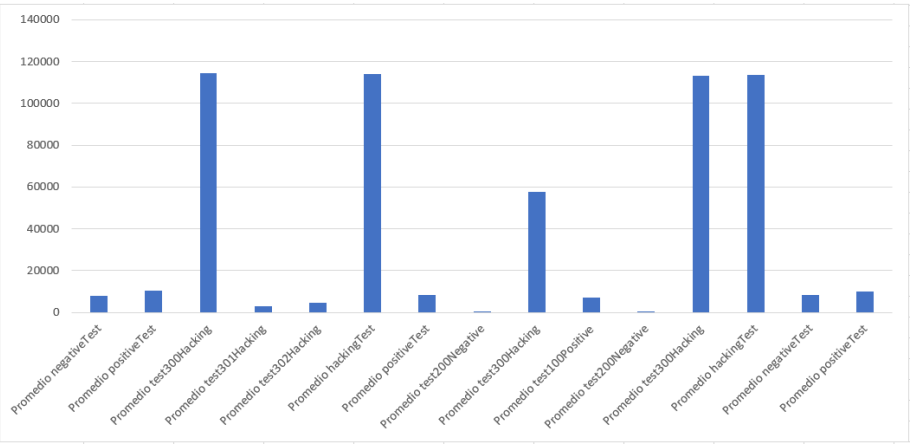
We have obtained the following graph of displaying each of the tests:

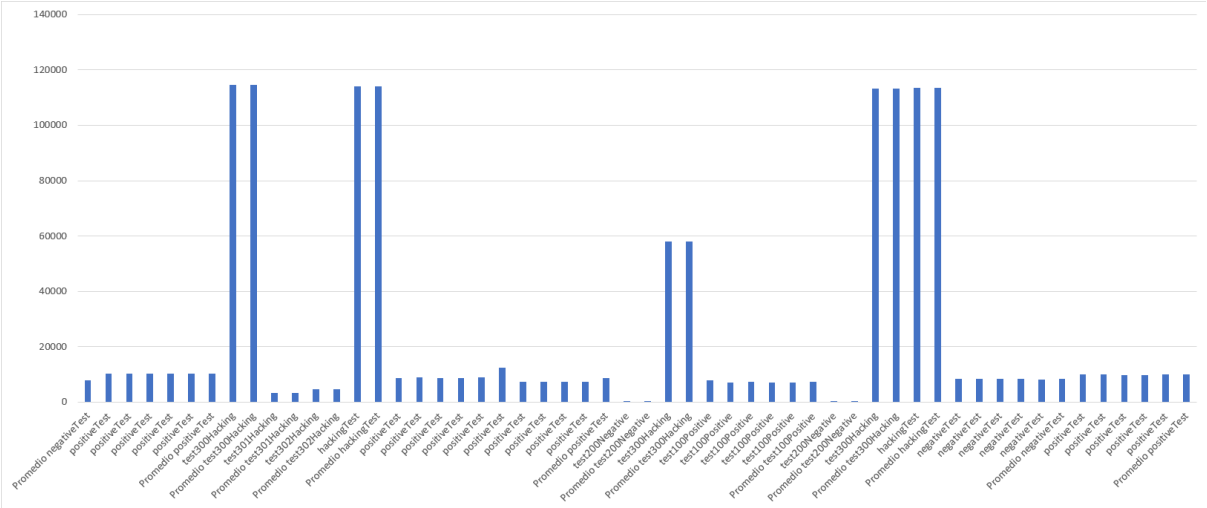


Intuitively, the conclusion is that most test cases take anywhere from about zero seconds to 90 seconds to complete.

Finally, graph your log. Intuitively, the conclusion is that most test cases take from roughly zero seconds up to 120 seconds to complete.

Promedio negativeTest	7848,2
Promedio positiveTest	10287,8
Promedio test300Hacking	114574
Promedio test301Hacking	3197
Promedio test302Hacking	4606
Promedio hackingTest	113943
Promedio positiveTest	8567,1
Promedio test200Negative	1
Promedio test300Hacking	57879
Promedio test100Positive	7238,2
Promedio test200Negative	1
Promedio test300Hacking	113300
Promedio hackingTest	113439
Promedio negativeTest	8252,6
Promedio positiveTest	9879,2
Promedio general	18730,4773





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

Columna1					
Media	7,4137931		Interval(ms)	6,59075548	8,23683073
Error típico	0,41857319		Interval(s)	0,00659076	0,00823683
Mediana	6				
Moda	2				
Desviación es	8,12722118				
Varianza de la	66,0517241				
Curtosis	32,772495				
Coeficiente de	4,56622075				
Rango	87				
Mínimo	1				
Máximo	88				
Suma	2795				
Cuenta	377				
Nivel de confi	0,82303762				

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.