

CARRERA:  
INGENIERIA EN SISTEMAS COMPUTACIONALES

ASIGNATURA:  
PROCESO PERSONAL DE DESARROLLO DE SOFTWARE

NOMBRE DE LA PRÁCTICA:  
REPORTE INTERMEDIO

INTEGRANTES DEL EQUIPO:  
GONZALO MARTINEZ SILVERIO

DOCENTE:  
DRA. TANIA TURRUBIATES LOPEZ

SEMESTRE:  
7

GRUPO:  
701<sup>a</sup>

## INTRODUCCIÓN

En este informe intermedio de PSP mostraremos una línea base del proceso que permita medir el progreso y la mejora del proceso personal, tanto durante el curso de PSP como en proyectos futuros guiados por PSP. Este informe proporcionará una referencia de los datos del proceso, lo que permitirá evaluar los siguientes aspectos:

- Analizar los datos relacionados con el tamaño y el tiempo.
- Entender la precisión de las estimaciones y los planes.
- Identificar en qué etapas del desarrollo se invierte más tiempo.
- Comprender cómo los defectos impactan en el tiempo de desarrollo y establecer estrategias para mejorar mi productividad.
- Evaluar la eficacia de las técnicas y herramientas que estoy utilizando en mi proceso.
- Identificar las áreas de mi proceso que requieren más atención y posiblemente una revisión.
- Observar las tendencias en mi rendimiento a lo largo del tiempo, lo que puede proporcionar información valiosa para la planificación futura.
- Establecer metas realistas para la mejora continua basadas en los datos recopilados.
- Fomentar una cultura de autoevaluación y aprendizaje constante, lo que es esencial para el crecimiento profesional.

En resumen, el informe intermedio de PSP es una herramienta invaluable para entender, evaluar y mejorar el proceso personal de desarrollo de software.

Fase #	Propósito	Guiar el análisis y redacción del informe intermedio PSP
	Criterio de entrada	Programas de 1 a 4 completados y revisados por el docente. Registro de tiempo y PSP forma resumida informe provisional. Registro de defectos en forma resumida.
1	Planeación	Estimación de tamaño del informe. <ul style="list-style-type: none"> <li>• Análisis del número de párrafos.</li> <li>• Número de tablas de datos / gráficos para crear.</li> </ul> Esfuerzo estimado en base al tamaño informe. Registro de estimaciones en el formato de resumen del plan. El tiempo de planificación registrándolo en la bitácora del registro del tiempo.
2	Desarrollo	Para cada pregunta análisis. <ul style="list-style-type: none"> <li>• Generar cuadro de análisis de datos o tabla.</li> <li>• Analizar gráfico / tabla y otros datos de proceso.</li> <li>• Análisis de escritura párrafo.</li> </ul> Registrar el tiempo de desarrollo en el formato de registro de tiempo.
3	Post mortem	Medir el tamaño real del informe. <ul style="list-style-type: none"> <li>• Número de tabla / tablas.</li> <li>• Número de párrafos de análisis.</li> </ul> Formulario de resumen del plan completo. Registrar tiempo post mortem en el formato de registro de tiempo.
	Criterio de salida	Reporte intermedio de PSP finalizado Resumen del plan finalizado Bitácora de tiempo finalizado

## RESUMEN DEL PLAN DEL REPORTE INTERMEDIO

Estudiante: Gonzalo Martínez Silverio

Fecha: 15/11/2023

Instructor: Tania Turrubiates Lopez

Size Data			Effort Estimate	
Objeto	Número Planeado	Número Real	Estimado por Objeto	Esfuerzo Estimado
Gráficas	20	15	30	600
Párrafos	30	22	20	600
Tabla	20	16	20	400
TOTAL	70	53	Total	1600

### Datos de Esfuerzo

FASE	TIEMPO PLAN	TIEMPO REAL
Planeación	10	12
Desarrollo	1600	870
Postmortem	120	180
TOTAL	1730	1062

## BITÁCORA DE TIEMPO DEL INFORME INTERMEDIO DE PSP

Estudiante: Gonzalo Martínez Silverio

Fecha: 15/11/2023

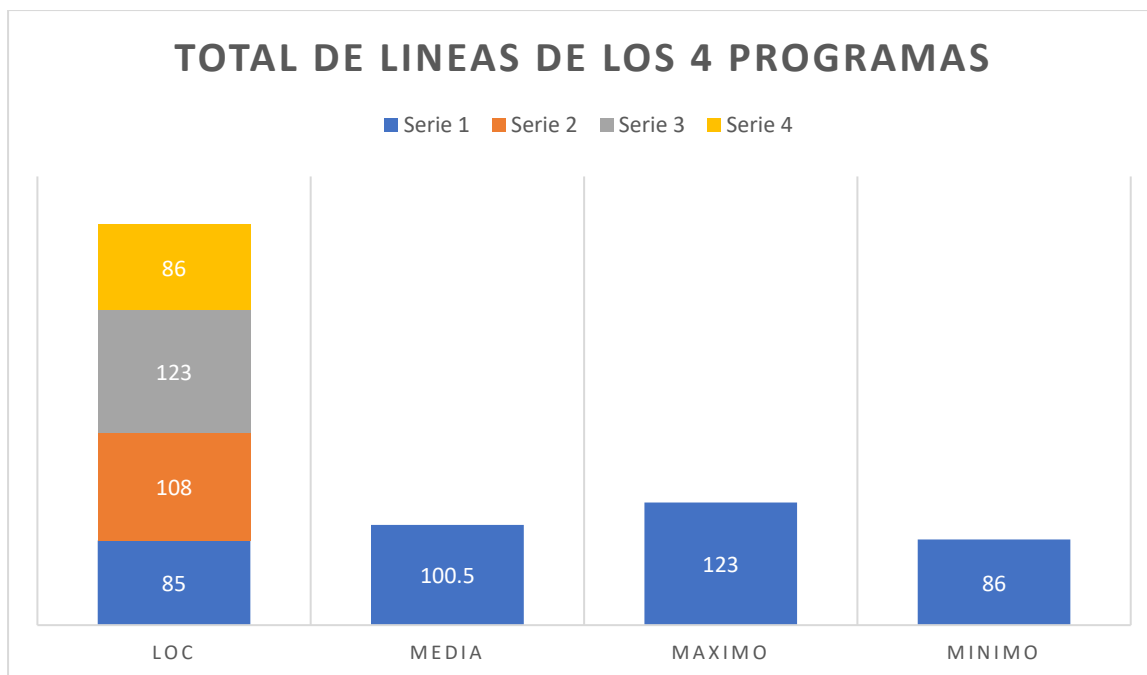
[illegible]

## DESARROLLO

### ANÁLISIS DE PRESICIÓN DE LA ESTIMACIÓN DEL TAMAÑO

1. ¿Cuáles son la media, máximo y tamaños reales mínimos de sus programas en LOC hasta la fecha?

LINEAS TOTALES DE EN LOS 4 PROGRAMAS				
No. Programa	LOC	Media	Máximo	Mínimo
1	85	100.5	123	86
2	108			
3	123			
4	86			

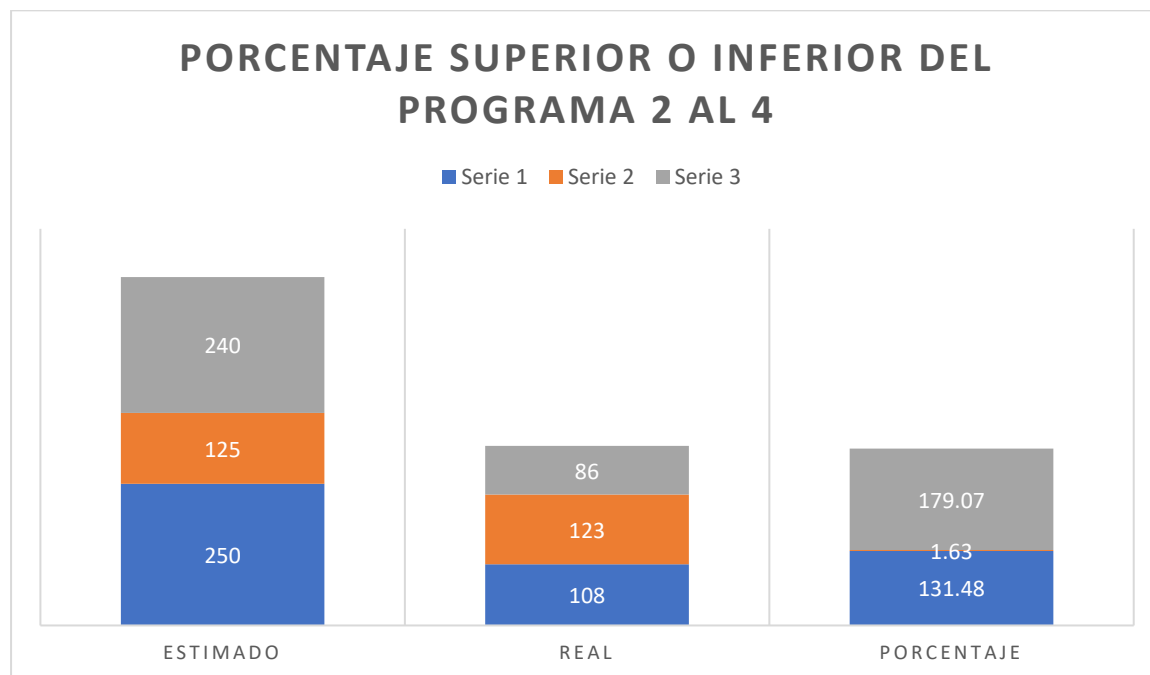


Contar los números de líneas en los 4 programas programa me permite medir mi habilidad de optimización de código.

En la gráfica anterior muestra el número de líneas en cada uno de los programas realizados como se puede observar el programa dos fue donde utilizamos más líneas, considero que fue por la complejidad del programa, además de no contar con líneas base, pero aun así se podría reducir las líneas buscando herramientas que puedan optimizar y simplificar sus procesos.

2. Excluyendo programa 1, ¿qué porcentaje superior o inferior al tamaño real fue el tamaño estimado (por ejemplo, si estimación/real se encuentra en %, el 85% es de 15% bajo, 120% es un 20% más) para cada programa?

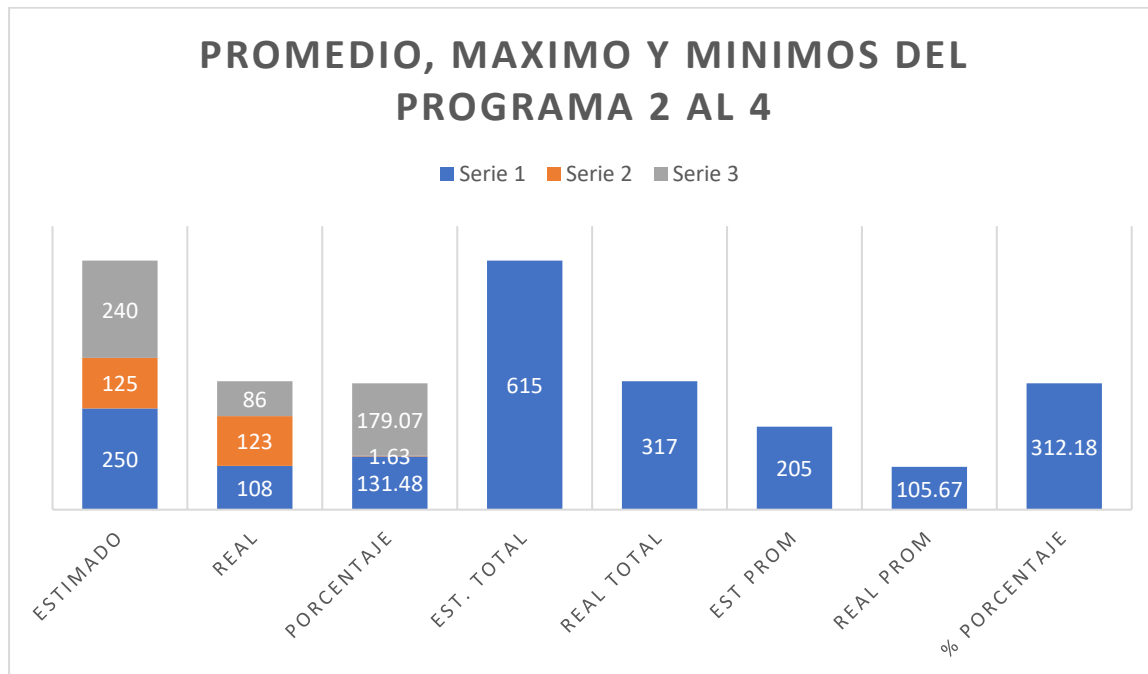
PORCENTAJE SUPERIOR O INFERIOR DEL PROGRAMA 2 AL 4			
No. Programa	Estimado	Real	Porcentaje
2	250	108	131.48
3	125	123	1.63
4	240	86	179.07



En la gráfica se muestra que tan certero fui e la hora de proyectar cuantas líneas necesitaba en cada programa, como se puede observar solo en el programa 3 pude acercarme a lo real ya que en el 2 y en el 4 tuve una diferencia superior del 131.480% y 179.07%, es decir que se sobreestimo, considero que esto sucedió porque ya había realizado un programa similar al 3 y por lo tanto pude estimar casi certeramente cuantas líneas iba a necesitar. Estimar cuantas líneas utilizará en un programa depende de la experiencia del programador.

3. ¿Cuáles son el promedio, máximo y valores mínimos para estos? Excluyendo programa 1, ¿cuál es el error total acumulado de las estimaciones del tamaño?

PROMEDIO, MAXIMO Y MINIMOS DEL PROGRAMA 2 AL 4								
No. Programa	Estimado	Real	Porcentaje	Est Total	Real total	Est prom	Real prom	% porcentaje
2	250	108	131.48	615	317	205	105.67	312.18
3	125	123	1.63					
4	240	86	179.07					



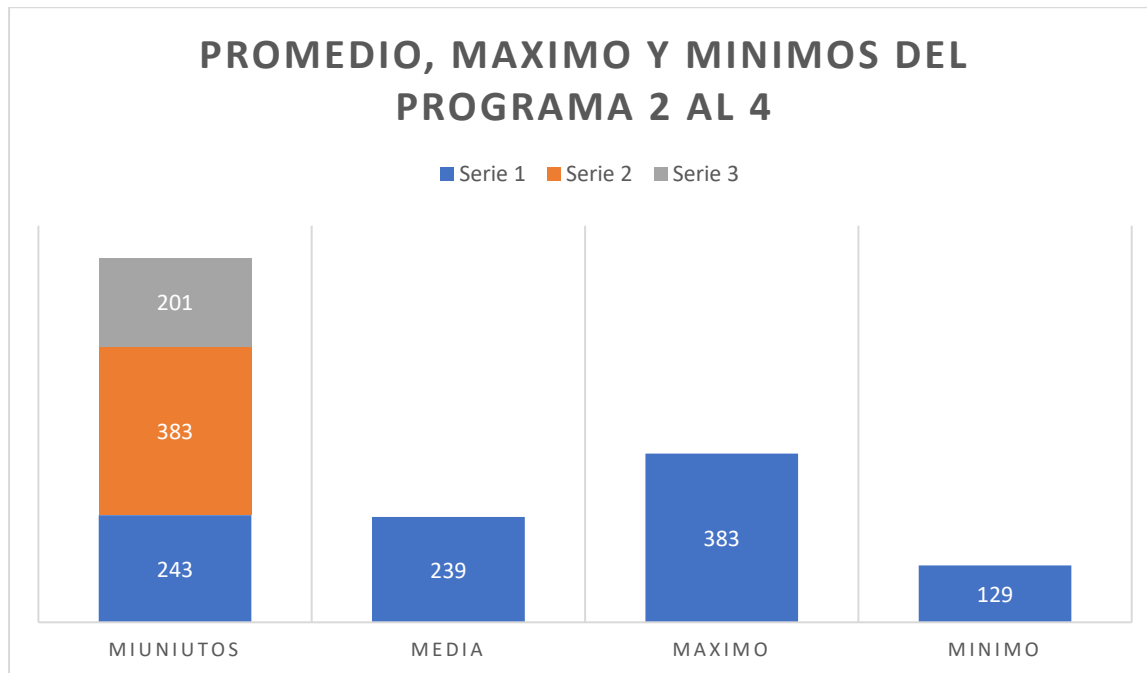
La gráfica muestra el total de líneas programadas esto nos arroja que tenemos un promedio de líneas reales 105.67 líneas, el programa donde menos líneas se agregaron o cambiaron fue en el 4 y el 2 en el que más se agregaron o cambiaron, pienso que esto sucedió porque a la hora de programar el programa 4 ya se tenían más líneas base que se utilizaron para realizarlo y en el programa 2 no se tenía ninguna base. Se sobreestimo en un 312.18% por lo tanto este es el error total que se obtuvieron al termino de los programas del 2 al 4.

#### ANÁLISIS DE PRESICIÓN DE ESTIMACIÓN DE TIEMPO Y PRODUCTIVIDAD.

¿Cuáles son el promedio, máximo y mínimo de tiempo en sus asignaciones de programa a la fecha?



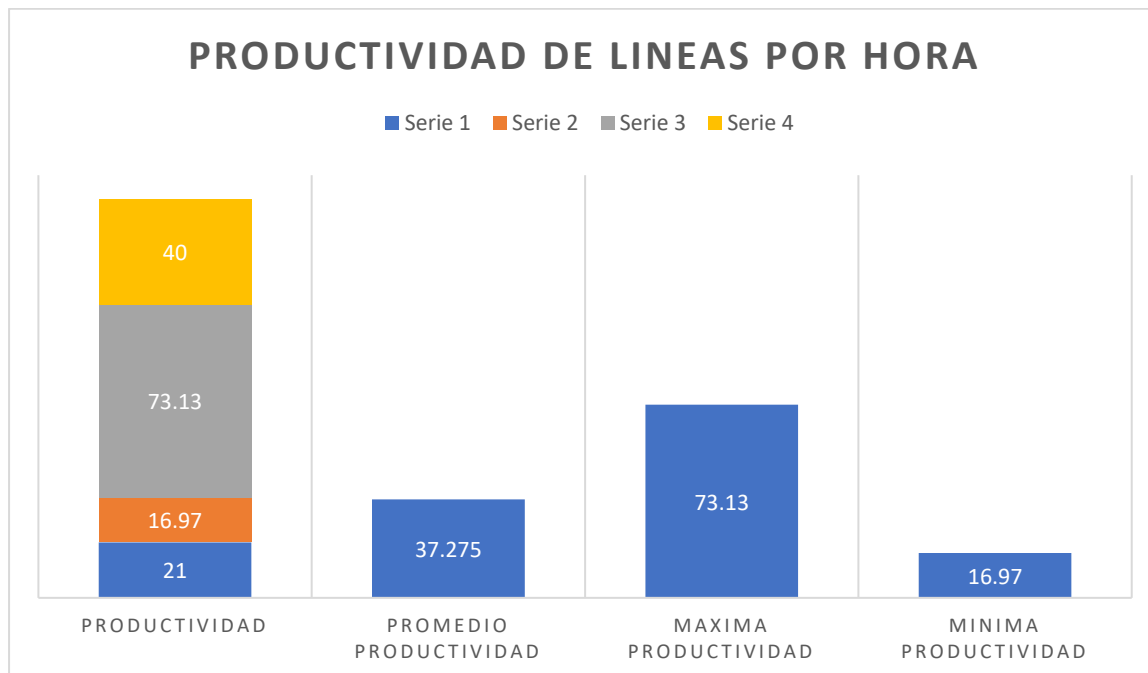
RESUMEN DEL TIEMPO EN LOS 4 PROGRAMAS				
No. Programa	MINUTOS	MEDIA	MÁXIMO	MÍNIMO
1	243	239	383	129
2	383			
3	201			
4	129			



El medir cuanto tiempo tardo en realizar un programa me permite saber que tan productivo soy a la hora de programar, en la gráfica anterior se muestra el resumen del tiempo que utilice en cada uno de los programas, además de los valores máximos, mínimos y la media de los mismos como se puede observar en el programa que más tarde fue en el dos ya que fue en el que se utilizaron más líneas de código y no se contaba con ninguna línea base, en el programa cuatro fue en el que menos tiempo se necesitó ya que en esa etapa del proceso se contaba con más experiencia en cuanto al mecanismo del mismo además de contar con líneas base para codificarlo.

¿Cuáles son la media, máximo y los valores mínimos de la productividad por programa hasta la fecha en LOC / HR?

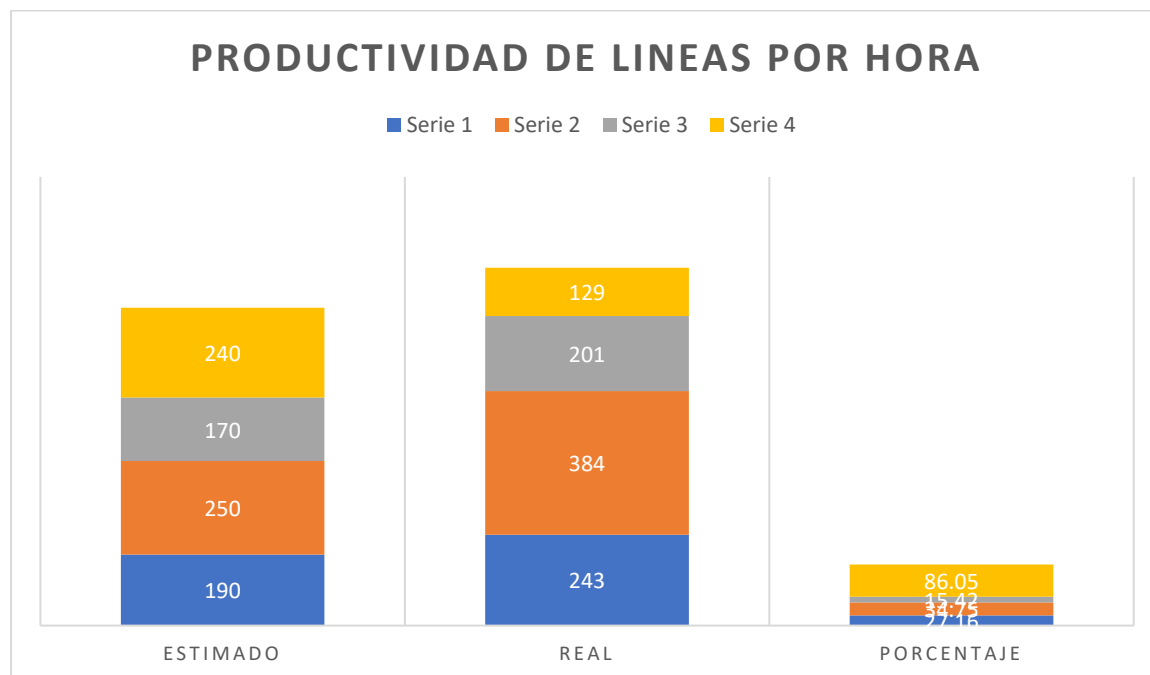
PRODUCTIVIDAD DE LINEAS POR HORA						
No. Programa	MINUTOS	LOC	PRODUCTIVIDAD	PROMEDIO PRODUCTIVIDAD	MÁXIMA PRODUCTIVIDAD	MÍNIMA PRODUCTIVIDAD
	243	85	21	37.275	73.13	16.97
	383	108	16.97			
	201	123	73.13			
	129	86	40			



En la gráfica anterior se muestra la productividad que obtuve en los cuatro programas donde en el programa 3 fue donde obtuve mayor productividad que se lo atribuyo a que la mayoría de los métodos creados contienen líneas similares y esto acelero la programación donde obtuve menor productividad es en el programa uno, creo que esto ocurre a consecuencia a la falta de experiencia en la utilización de funciones del lenguaje empleado en los programas en este caso JAVA

¿Qué porcentaje mayor o menor al tiempo real fue el tiempo estimado (por ejemplo, si estimado / real es en%, el 85% es de 15% bajo, 120% es un 20% más) para cada programa?

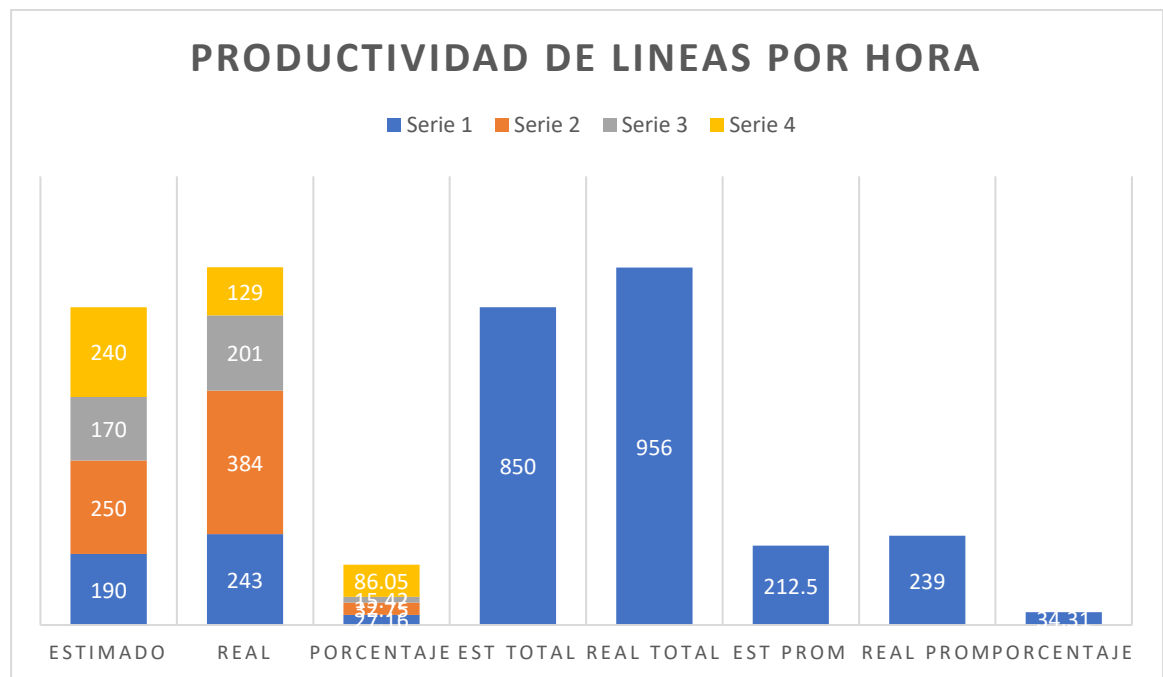
PORCENTAJE SUPERIOR O INFERIOR SOBRE EL TIEMPO			
No. Programa	Estimado	Real	Porcentaje
1	190	243	27.16% mayor
2	250	383	32.75% menor
3	170	201	15.43% menor
4	240	129	86.05% mayor



En la gráfica anterior indican cuánto porcentaje el tiempo estimado es mayor o menor que el tiempo real para cada programa. Un valor negativo indica que el tiempo estimado es menor que el tiempo real, y un valor positivo indica que el tiempo estimado es mayor que el tiempo real. Se puede observar en el programa 4 fue donde se sobreestimo más con un 86.05% esto debido a que no se contaba con experiencia y fue la primera vez que, hacia estos cálculos, en donde se subestimo más es en el programa cuatro ya que a la hora de estimar se tenía claro cuantas líneas base se podrían utilizar.

¿Cuáles son el promedio, máximo y valores mínimos para estos? ¿cuál es el error total acumulado de las estimaciones de tiempo (es decir, el tiempo estimado total / tiempo real total)?

No. Programa	Estimado	Real	Porcentaje	ESTIMADO TOTAL	REAL TOTAL	ESTIMADO PROMEDIO	REAL PROMEDIO	% PORCENTAJE
1	190	243	27.16	850	956	212.5	239	34.31
2	250	383	32.75					
3	170	201	15.43					
4	240	129	86.05					

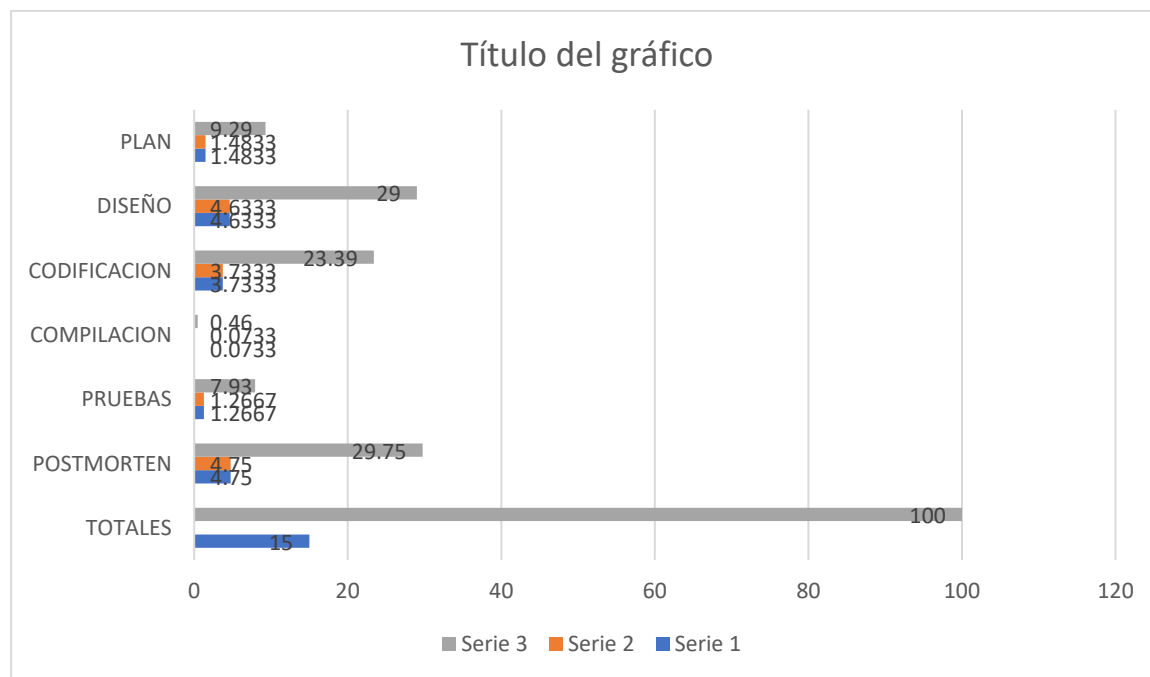


La gráfica muestra el total de tiempo que tarde en programar todos los programas esto nos arroja que tengo un promedio de tiempo de 239 minutos (3.9 horas) por programa, el programa donde menos tiempo tarde en programar fue en el programa 4 esto creo porque código que se es muy similar en todos los métodos empleados, en el programa 3 fue donde tarde más en programar esto lo atribuyo a que no contaba con base. Obtuve un error total de 34.31% sobrestimando el tiempo que estimado contra el real.

## ANÁLISIS DE LA DISTRIBUCIÓN DEL TIEMPO.

¿Qué porcentaje de tiempo dedica en promedio en cada fase del proceso (planificación, diseño, código, compilación, pruebas, postmortem)? ¿en el diseño y el código en su conjunto? ¿en la compilación y la prueba en su conjunto?

PROMEDIO DE TIEMPO EN HORAS EN CADA FASE			
FASE	TOTAL EN CADA FASE	PROMEDIO EN CADA FASE	PORCENTAJE
PLAN	1.4833	1.4833	9.29
DISEÑO	4.6333	4.6333	29
CODIFICACIÓN	3.7333	3.7333	23.39
COMPILACIÓN	0.0733	0.0733	0.46
PRUEBAS	1.2667	1.2667	7.93
POSTMORTEM	4.75	4.75	29.75
TOTALES	15.94	-	100
DISEÑO Y CÓDIGO	8.3667	4.18335	52.44
COMPILACIÓN Y PRUEBA	1.34	0.67	8.39



En esta gráfica se muestra el tiempo dedicado en cada fase de PSP donde se observa que la fase que más le dedico tiempo es a la de postmortem ya que se requiere de mucho análisis y un grado de detalle muy alto a la que menos le dedico tiempo es a la fase de compilación ya que esta fase es muy rápida y casi no tuvo errores. En las fases de diseño y código se tiene un 52.44% del total de proceso yo considero que estas son las importantes del proceso ya que diseñando correctamente las fases restantes son muy fáciles de realizar ya que no se obtienen errores y por lo tanto las pruebas son más

rápidas. Se obtuvo un 8.39% en las fases de compilación y pruebas, en estas fases comprobamos si el diseño y el código están correctos ya que en es donde se detectan los errores.

**SI VAS A GASTAR 1000 HORAS ESCRIBIENDO PROGRAMAS SIMILARES A LOS DE LA CLASE,**

**¿Cuánto tiempo es probable que pasen en el diseño?**

Considerando que en la gráfica anterior se obtuvo el promedio de la etapa de diseño que es del 29%, entonces se multiplica y las 1000 son el 100% del proceso, se realiza la siguiente operación  $(29.24 * 1000) / 100$  que da un resultado de 290 horas, estas son las horas que posiblemente se le dedique a la etapa de diseño.

**¿En la codificación?**

Se aplica el mismo procedimiento se multiplica  $(23.39 * 1000) / 100$  que da un resultado de 233.9 horas que es el tiempo posible que se dedicara a la fase de codificación.

**¿En pruebas?**

En las pruebas se multiplica  $(7.93 * 1000) / 100$  que da un resultado de 79.3 horas que son la que posiblemente se dedicaran a la fase de pruebas.

**¿Cuántas líneas de código es probable que escriba en ese momento?**

Teniendo en cuenta que tengo un promedio de productividad de 37.275 líneas por hora en todo el proceso de mis 4 programas entonces solo hay que multiplicar el total del tiempo que son 1000 horas por las líneas de la productividad, entonces la operación queda la siguiente forma  $1000 * 37.275$  y el resultado es el siguiente 37,275 líneas en todo el proceso.

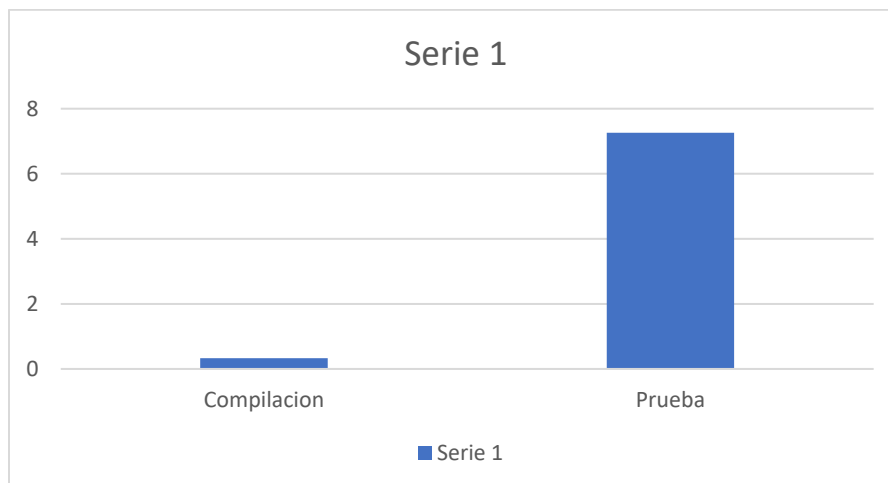
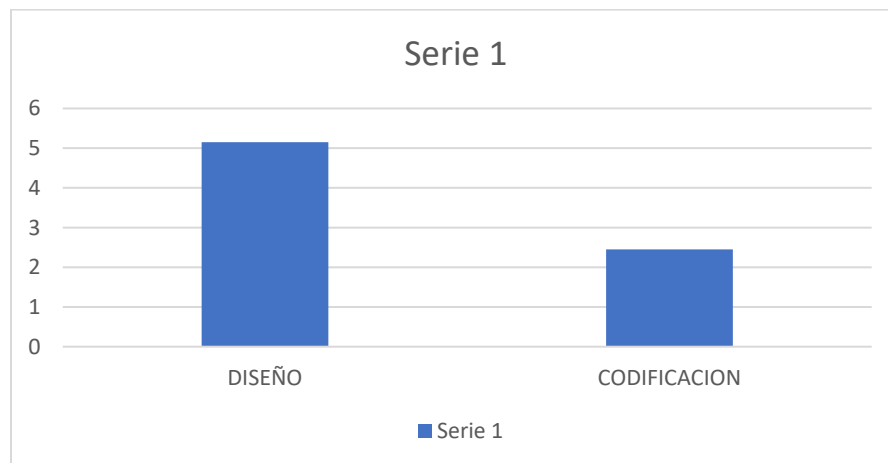
**¿Cuántos defectos es probable que encuentre en pruebas?**

Es posible que se encuentren dos errores ya que es el promedio que tengo en mis cuatro programas

## ANALISIS DEL TIEMPO DE REPARACIÓN DE DEFECTOS

**Analizar los tiempos de reparación de defectos, en base a la fase en que fueron inyectados y removidos.**

TIEMPO DE ERRORES POR FASE INYECTADA Y REMOVIDA			
FASE QUE SE INYECTA	TIEMPO	FASE QUE REMUEVE	TIEMPO
DISEÑO	5.15	COMPILACION	0.33
CODIFICACION	2.45	PRUEBA	7.26

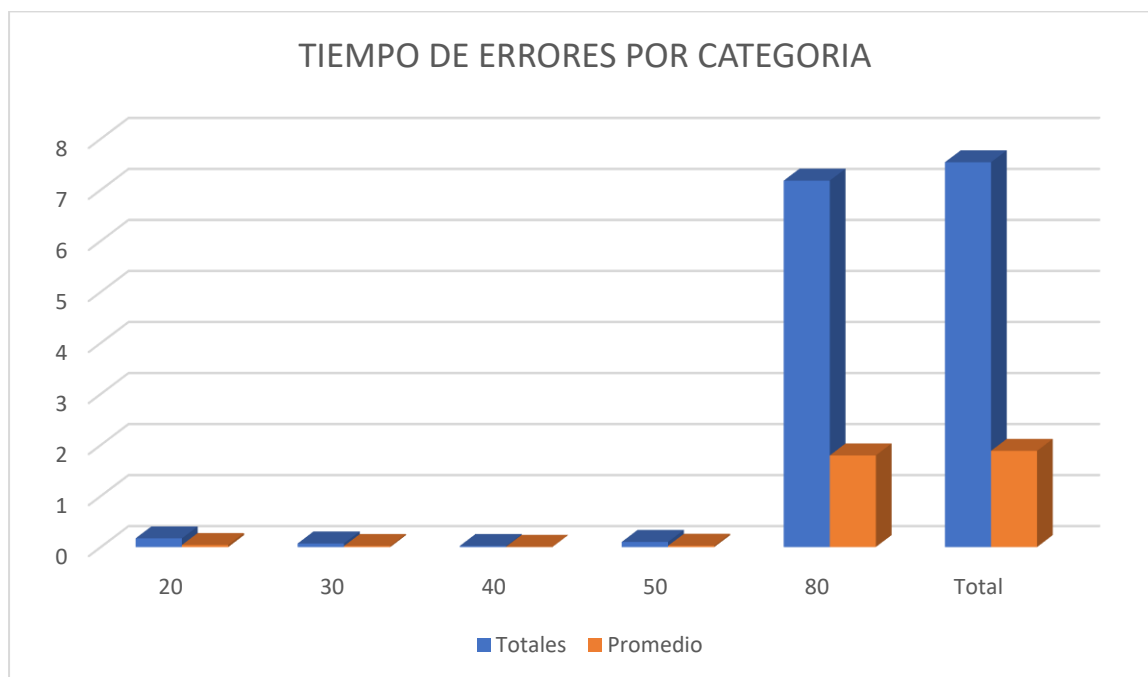


La fase en la que obtuve mayor tiempo inyectando errores es la de diseño con un total de 5.15 horas esto lo atribuyo a que no hice correctamente el diseño esto me servirá para hacer un mejor diseño para tener menos errores a la hora de codificar.

Donde más se detectaron los errores y en su defecto se corrigieron fue en la etapa de prueba, esto es porque en esta fase es donde se comprueba el funcionamiento del sistema y se espera que los resultados sean los esperados.

¿Qué categoría en promedio ha tenido el mayor tiempo de reparación? ¿qué categoría ha tenido el mayor tiempo de reparación?

TIEMPO DE ERRORES POR CATEGORIA						
CATEGORIAS						
PROGRAMA	20	30	40	50	80	Total
1	0.08	0.03	0.01	0.1	2.11	2.33
2	0.08	0.01	0.01	0	5.06	5.16
3	0.01	0.03	0	0	0	0.04
4	0	0	0	0	0.01	0.01
Totales	0.17	0.07	0.2	0.1	7.18	7.54
Promedio	0.04	0.02	0.005	0.025	1.795	1.885



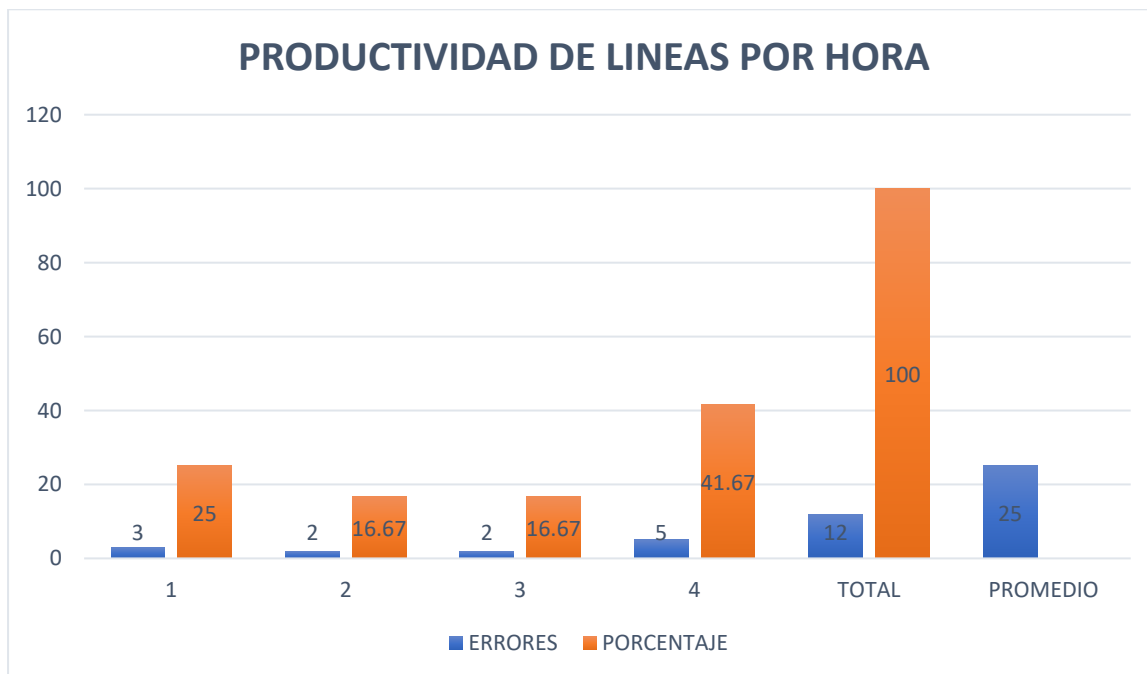
La categoría de interfaz que es el error número 80 fue la que en promedio y en total obtuvo el mayor tiempo de reparación, pienso que esto sucede porque no se realizó un buen diseño y a la hora de programar se introducen errores.



## ANÁLISIS DE DEFECTOS.

¿Cuáles son la media, máximo y valores mínimos para el total de defectos por programa hasta la fecha?

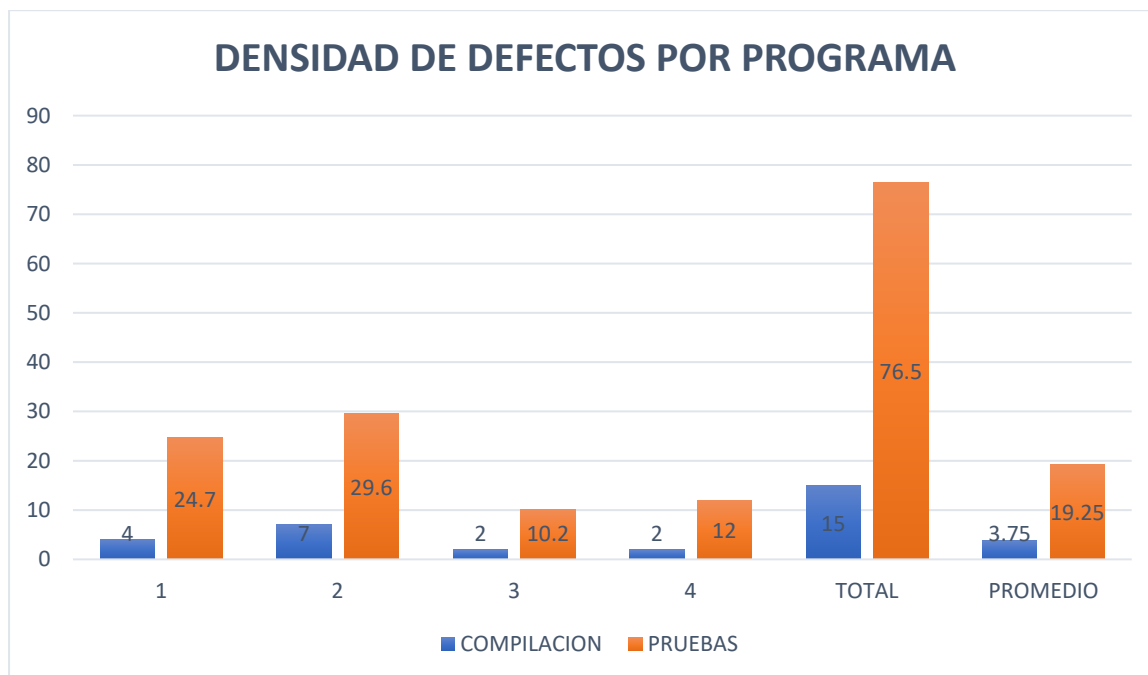
ERRORES POR PROGRAMA		
NO. PROGRAMA	ERRORES	PORCENTAJE
1	3	25
2	2	16.67
3	2	16.67
4	5	41.67
TOTAL	12	100
PROMEDIO	3	



En el programa cuatro es donde obtuve más errores con 5 en total, el dos y el tres fue donde menos errores obtuve con un total de 2 errores, esto es porque ya contaba con más experiencia al realizar los procesos, en promedio se generaron 3 errores por programa

¿Cuáles son la media, máximo y los valores mínimos de densidad de defectos en la compilación y pruebas por programa hasta la fecha?

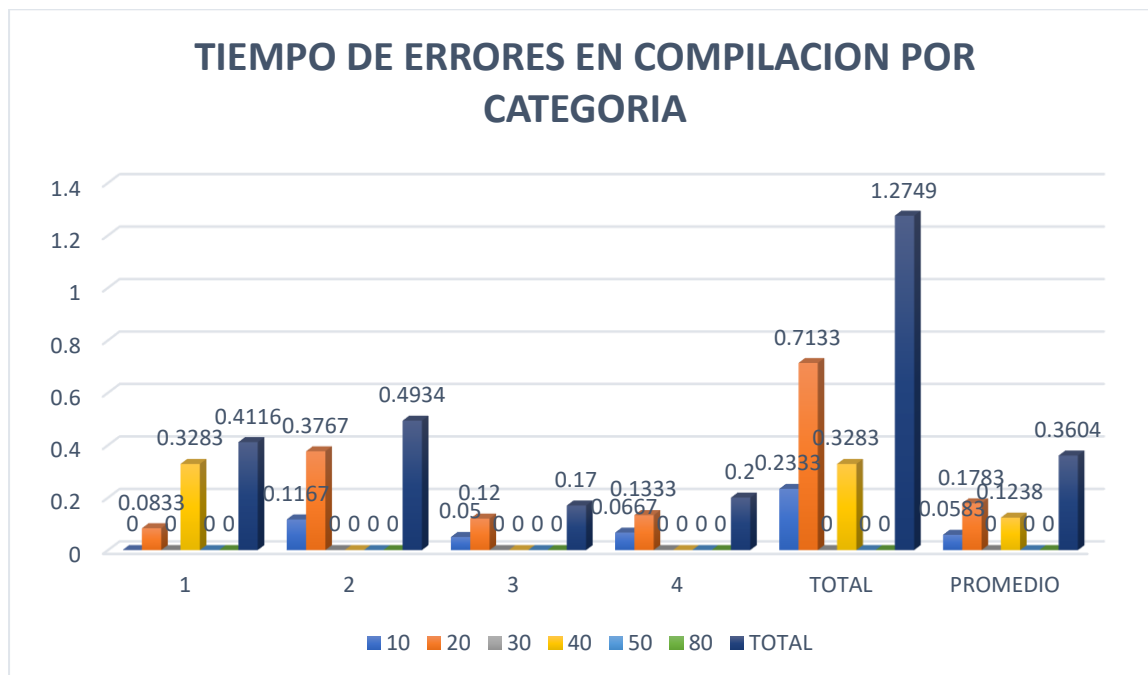
DENSIDAD DE DEFECTOS POR PROGRAMA		
NO. PROGRAMA	COMPILACION	PRUEBAS
1	4	24.7
2	7	29.6
3	2	10.2
4	2	12.0
TOTAL	15	76.5
PROMEDIO	3.75	19.25



En la gráfica se muestran los máximos, mínimos y el promedio de la densidad de defectos encontrados en las fases de compilación y prueba, en el programa dos fue donde se encuentra la mayor densidad de defectos en la fase de compilación, considero que esto paso porque no realice un buen diseño. Además, también fue donde obtuve la mayor densidad en la fase de pruebas, pienso que esto fue por la complejidad del programa. En promedio en la fase de compilación obtuve densidad de 3.75 defectos por programa y en las pruebas 2 defectos por programa.

¿Qué tipo de defecto ha durado más tiempo en la fase compilación?

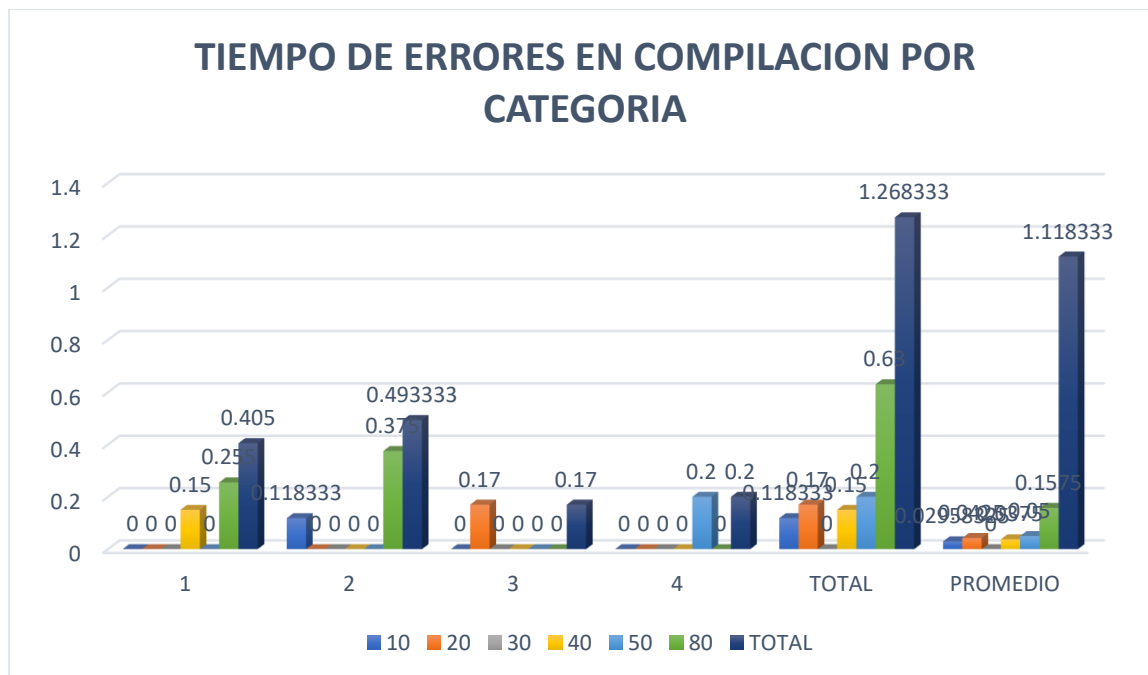
TIEMPO DE ERRORES EN COMPILACION POR CATEGORIA							
NO. PROGRAMA	10	20	30	40	50	80	TOTAL
1	0	0.0833	0	0.3283	0	0	0.4117
2	0.1167	0.3767	0	0	0	0	0.4933
3	0.05	0.12	0	0	0	0	0.17
4	0.0667	0.1333	0	0	0	0	0.2
TOTAL	0.2333	0.7133	0	0.3283	0	0	1.275
PROMEDIO	0.0583	0.1783	0	0.1238	0	0	0.3604



El defecto de tipo 20 que corresponde a sintaxis es el que más ha durado en la fase de compilación con 0.7133 horas (42.8 minutos), lo atribuyo a que no se escribo el código correctamente y no respetar el diseño del mismo.

¿Y EN PRUEBAS?

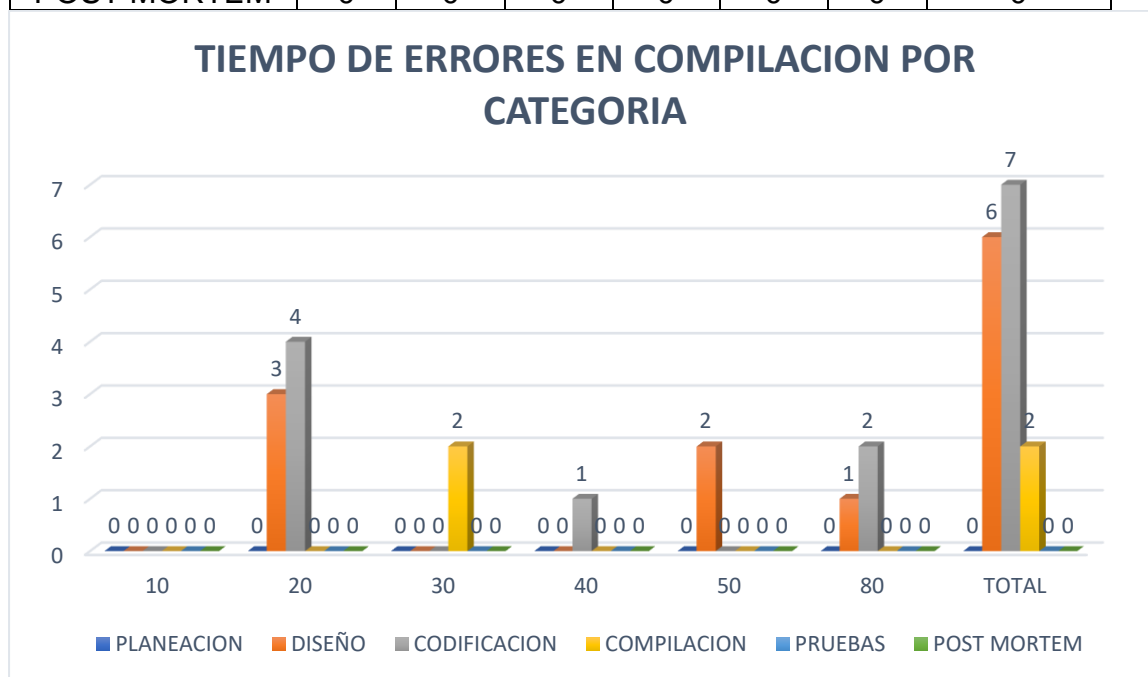
TIEMPO DE ERRORES EN PRUEBAS POR CATEGORIA							
NO. PROGRAMA	10	20	30	40	50	80	TOTAL
1	0.00	0.00	0.00	0.15	0.00	0.255	0.405
2	0.118333	0.00	0.00	0.00	0.00	0.375	0.493333
3	0.00	0.17	0.00	0.00	0.00	0.00	0.17
4	0.00	0.00	0.00	0.00	0.2	0.00	0.21
TOTAL	0.118333	0.17	0.00	0.15	0.2	0.63	1.268333
PROMEDIO	0.02958325	0.0425	0	0.0375	0.05	0.1575	1.118333



El defecto de tipo 80 que corresponde a la interfaz es el que más ha durado en la fase de compilación con 0.375 horas, considero que esto sucedió a que desde un inicio no se diseñó bien el programa.

¿EN CUAL FASE SON INYECTADOS MÁS SEGUIDO DEFECTOS Y DE QUE TIPO SON?

DEFECTOS POR FASE Y TIPO								DEFECTOS POR FASE Y TIPO
NO. PROGRAMA	10	20	30	40	50	80	TOTAL	
PLANEACION	0	0	0	0	0	0	0	
DISEÑO	0	1	0	0	2	1	0	
CODIFICACION	0	8	0	1	0	2	0	
COMPILACION	0	0	0	0	0	0	0	
PRUEBAS	0	0	0	0	0	0	0	
POST MORTEM	0	0	0	0	0	0	0	



La fase en la cual tuve más errores fue en la de codificación, el tipo de errores que se inyectaron en dicha fase son los siguientes: 20, 40 y 80 que corresponden a sintaxis, paquetes, asignación y función respectivamente, nuevamente se lo atribuyo a que no diseñé bien y además que, dicha fase es la más compleja ya que se necesita tener mucha experiencia para poder detectar y eliminar estos tipos errores.



Student Gonzalo Martinez Silverio

Start Date 07-sep.-23

Program Program 1

End Date 12-sep.-23

Instructor TANIA TURRUBIATES LOPEZ

Language Java

## Time in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN		13	0	0.0%
DLD		59	0	0.0%
CODE		50	0	0.0%
COMPILE		0	0	0.0%
UT		24	0	0.0%
PM		96	0	0.0%
Total	190	243	0	

## Defects Injected in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN		0	0	0.0%
DLD		2	0	0.0%
CODE		3	0	0.0%
COMPILE		0	0	0.0%
UT		0	0	0.0%
PM		0	0	0.0%
Total		5	0	

## Defects Removed in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN		0	0	0.0%
DLD		0	0	0.0%
CODE		1	0	0.0%
COMPILE		1	0	0.0%
UT		3	0	0.0%
PM		0	0	0.0%
Total		5	0	

## ID 422

Start Date	24-sep.-23
------------	------------

End Date	26-sep.-23
----------	------------

Language Java

LOC-Lines of code 

	Plan Size	Actual Size	To Date
Base (B)		0.00	
Deleted (D)		0.00	
Modified (M)		67.00	
Added (A)		0.00	
Reused (R)		0.00	0.00
Added & Modified (A&M)	85	67.00	67.00
Total (T)		0.00	0.00
New Reusable (NR)		0.00	0.00

Phase	Plan	Actual	To-Date	To-Date%
-------	------	--------	---------	----------

PLAN	13	34	13	5.2%
DLD	61	140	59	24.3%
CODE	52	75	50	20.7%
COMPILE	0	1	0	0.2%
UT	25	30	24	10.0%
PM	99	104	96	39.6%

Total	250	383	243
-------	-----	-----	-----

Phase	Plan	Actual	To-Date	To-Date%
-------	------	--------	---------	----------

PLAN	0	0	0.0%
DLD	2	2	40.0%
CODE	2	3	60.0%
COMPILE	0	0	0.0%
UT	0	0	0.0%
PM	0	0	0.0%

Total	4	5
-------	---	---

Phase	Plan	Actual	To-Date	To-Date%
-------	------	--------	---------	----------

PLAN	0	0	0.0%
DLD	0	0	0.0%
CODE	2	1	20.0%
COMPILE	0	1	20.0%
UT	2	3	60.0%
PM	0	0	0.0%

Total	4	5
-------	---	---

PSP1 Project Plan Summary

Personal Software Process<sup>SM</sup>

Student

Gonzalo Martinez Silverio

Program

Program 3

Instructor

TANIA TURRUBIATES LOPEZ

Start Date

14-oct.-23

End Date

15-oct.-23

Language

Java

Summary

Plan

Actual

To-Date

Productivity

36.0

34.1

17.3

Program Size Summary

LOC-Lines of code

Plan Size

Actual Size

To Date

Base (B)

185.00

185.00

Deleted (D)

0.00

62.00

Modified (M)

0.00

123.00

Added (A)

102.00

-9.00

Reused (R)

4.00

9.00

9.00

Added & Modified (A&M)

102.00

114.00

181.00

Total (T)

291.00

123

123.00

New Reusable (NR)

0.00

0.00

0.00

Estimated A&M (E)

102.00

Time in Phase

Phase

Plan

Actual

To-Date

To-Date%

PLAN

13

22

47

7.4%

DLD

54

49

199

31.8%

CODE

34

62

125

20.0%

COMPILE

0

2

1

0.2%

UT

15

10

54

8.6%

PM

54

55

200

31.9%

Total

170

201

626

Defects Injected in Phase

Phase

Plan

Actual

To-Date

To-Date%

PLAN

0

0

0.0%

DLD

0

4

44.4%

CODE

2

5

55.6%

COMPILE

0

0

0.0%

UT

0

0

0.0%

PM

0

0

0.0%

Total

2

9

Defects Removed in Phase

Phase

Plan

Actual

To-Date

To-Date%

PLAN

0

0

0.0%

DLD

0

0

0.0%

CODE

2

3

33.3%

COMPILE

0

1

11.1%

UT

0

5

55.6%

PM

0

0

0.0%

Total

2

9

Registro: 1 de 1

Filtrado

Buscar



PSP1.1 Project Plan Summary

Personal Software Process<sup>SM</sup>

Student

Gonzalo Martinez Silverio

Program

Program 4

Instructor

TANIA TURRUBIATES LOPEZ

Start Date

03-nov.-23

End Date

03-nov.-23

Language

Java

ID

424

Summary

	Plan	Actual	To-Date
Productivity	14.5	23.7	16.8
Planned Time	240.0		850.0
Actual Time		129.4	956.3
CPI			0.9
%Reused	0.0	1.9	5.7
%New Reusable	0.0	0.0	0.0

Program Size Summary

LOC-Lines of code

	Plan Size	Actual Size	To Date
Base (B)	108.00	108.00	
Deleted (D)	0.00	58.00	
Modified (M)	0.00	50.00	
Added (A)	58.00	1.00	
Reused (R)	0.00	1.00	10.00
Added & Modified (A&M)	58.00	51.00	232.00
Total (T)	166.00	52	175.00
New Reusable (NR)	0.00	0.00	0.00
Estimated A&M (E)	58.00		

Time in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN	20	20	69	8.3%
DLD	72	30	249	30.1%
CODE	54	37	187	22.6%
COMPILE	1	0	3	0.4%
UT	19	12	64	7.7%
PM	74	30	255	30.9%
Total	240	129	827	

Defects Injected in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN		0	0	0.0%
DLD		1	4	36.4%
CODE		1	7	63.6%
COMPILE		0	0	0.0%
UT		0	0	0.0%
PM		0	0	0.0%
Total		2	11	

Defects Removed in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN		0	0	0.0%
DLD		0	0	0.0%
CODE		1	5	45.5%
COMPILE		0	1	9.1%
UT		1	5	45.5%
PM		0	0	0.0%
Total		2	11	

Registro: 1 de 1

Filtrado

Buscar