

Instituto Tecnológico Superior de Nochistlán



Ingeniería en Sistemas Computacionales

PROCESO PERSONAL PARA EL DESARROLLO DE SOFTWARE

MES. J Jesús Minero Guardado

REPORTE INTERMEDIO PSP

Jorge Luis Vallín Medel

Nochistlán de Mejía, Zacatecas. 16 de abril de 2015

INDICE

INTRODUCCIÓN.....	3
REPORTE INTERMEDIO	4
RESUMEN DEL PLAN DEL REPORTE INTERMEDIO.....	5
BITÁCORA DE REGISTRO DE TIEMPO DEL REPORTE INTERMEDIO DE PSP	6
DESARROLLO.....	7
ANÁLISIS DE PRESICIÓN DE LA ESTIMACIÓN DEL TAMAÑO	7
ANÁLISIS DE PRESICIÓN DE ESTIMACIÓN DE TIEMPO Y PRODUCTIVIDAD.	10
ANÁLISIS DE LA DISTRIBUCIÓN DEL TIEMPO.....	14
ANALISIS DEL TIEMPO DE REPARACIÓN DE DEFECTOS	16
ANÁLISIS DE DEFECTOS.	18
EVIDENCIA DEL REPORTE.....	23
PROGRAMA 1 SUMMARY	23
PROGRAMA 2 SUMMARY	24
PROGRAMA 3 SUMMARY	25
PROGRAMA 4 SUMMARY	26
TIME LOG.....	27
DEFECT LOG.....	28

INTRODUCCIÓN

El objetivo del reporte intermedio de PSP es establecer un línea de referencia del proceso contra la cual pueda medirse la mejora de mi proceso personal tanto y durante en el curso de PSP así como cuando use PSP para guiar proyectos posteriores.

Este reporte me servirá para brindarme una referencia de los datos de mi proceso con la cual pueda evaluar los siguientes puntos:

- Analizar los datos de tamaño y tiempo.
- Comprender la exactitud de los estimados y planes
- Comprender dónde se gasta más tiempo durante el desarrollo
- Comprender cómo los defectos afectan su tiempo de desarrollo y poder establecer estrategias para mejorar mi productividad.

REPORTE INTERMEDIO

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. -Análisis del número de párrafos. -Número de tablas de datos / gráficos para crear. 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. - Generar cuadro de análisis de datos o tabla. - Analizar gráfico / tabla y otros datos de proceso. - Análisis de escritura párrafo. Registrar el tiempo de desarrollo en el formato de registro de tiempo.
3	POST MORTEM	<ul style="list-style-type: none"> • Registrar tamaño real del informe. - Número de tabla / tablas. - Número de párrafos de análisis. • Formulario de Resumen del Plan Completo. • Registrar tiempo postmortem 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

Jorge Luis Vallín Medel

Fecha

16 de abril de 2015

Instructor

J. Jesús Minero Guardado

Datos de Tamaño			Esfuerzo Estimado	
Objeto	Número Planeado	Número Real	Esf. Estimado por Objeto	Esfuerzo Estimado
Gráficas	20	15	30	600
Parrafos	30	22	20	600
Tablas	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 REGISTRO DE TIEMPO DEL REPORTE INTERMEDIO DE PSP

Estudiante: Jorge Luis Vallín Medel

Fecha: 16 de abril de 2015

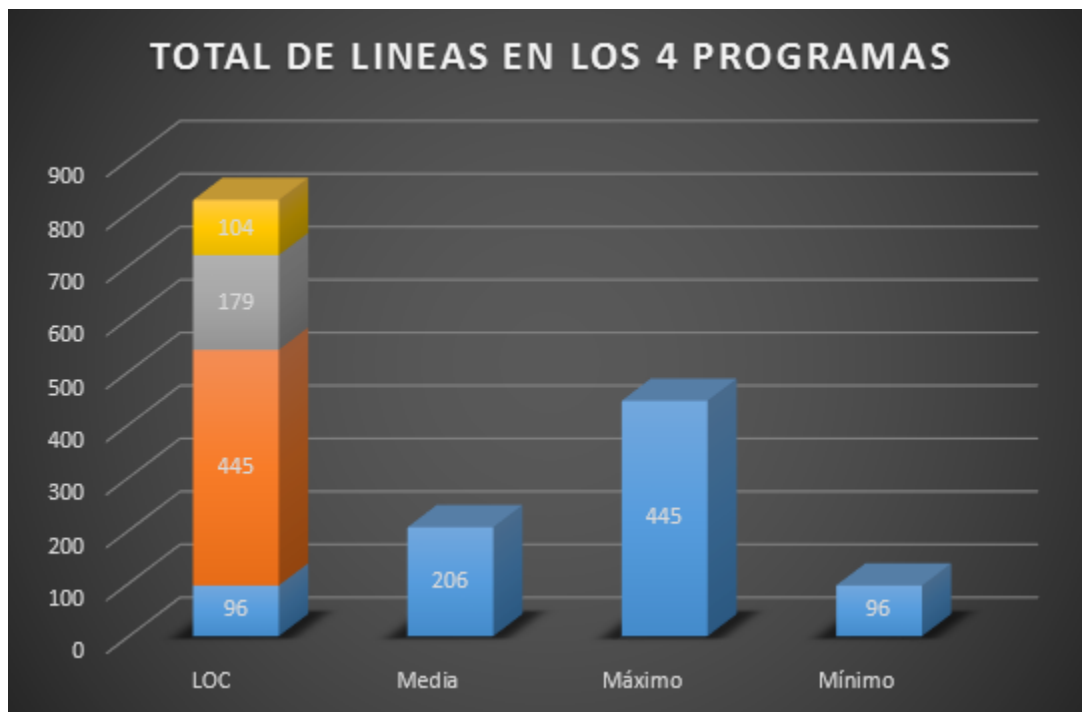
[illegible]

DESARROLLO

ANÁLISIS DE PRESICIÓN DE LA ESTIMACIÓN DEL TAMAÑO

¿CUÁLES SON LA MEDIA, MÁXIMO Y TAMAÑOS REALES MÍNIMOS DE SUS PROGRAMAS EN LOC HASTA LA FECHA?

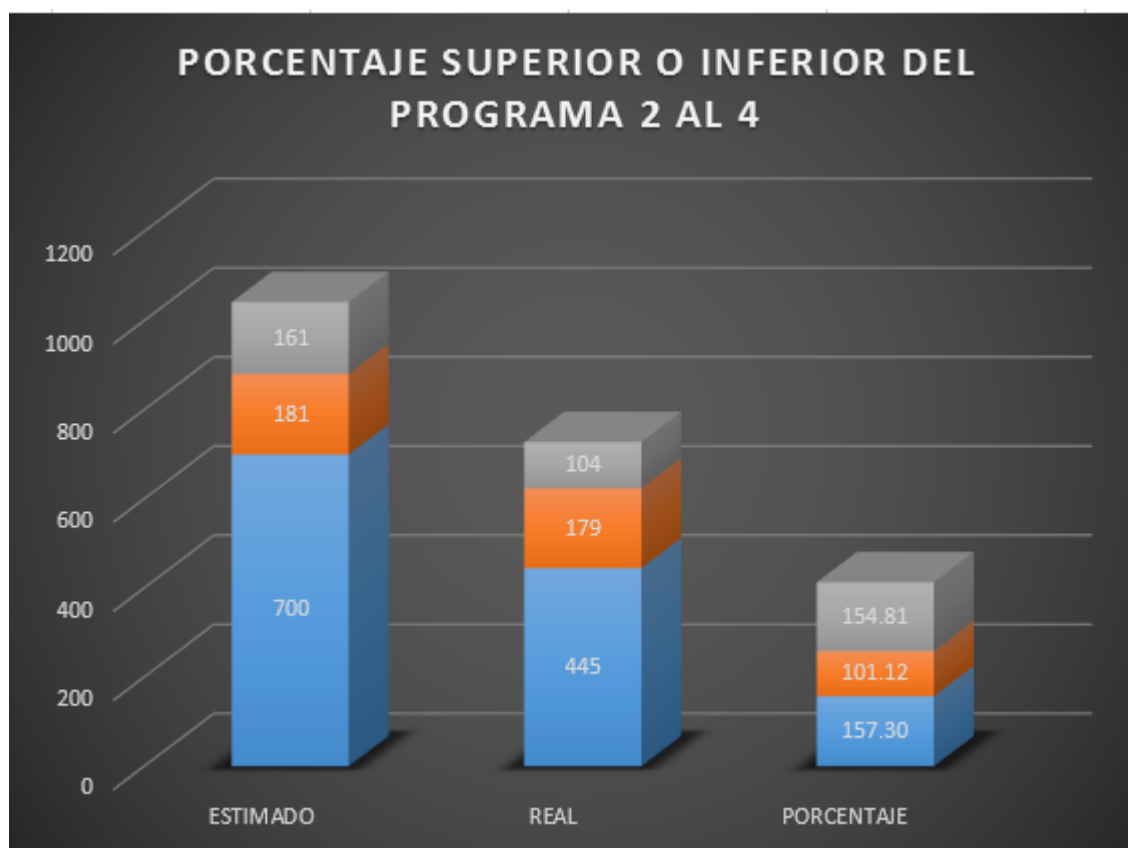
TOTAL DE LINEAS EN LOS 4 PROGRAMAS				
No. Programa	LOC	Media	Máximo	Mínimo
1	96	206	445	96
2	445			
3	179			
4	104			



El contar el número de líneas empleadas en cada programa me permite medir mi habilidad de optimización de código, en la gráfica anterior se muestra el número de líneas en cada uno de los programas realizados como se puede observar el programa dos fue donde utilice más líneas, considero que fue por la complejidad del programa, además de no contar con líneas base, pero aun así se podría reducir la líneas buscando herramientas que puedan optimizar y simplificar los procesos de dicho programa.

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	700	445	157.30
3	181	179	101.12
4	161	104	154.81

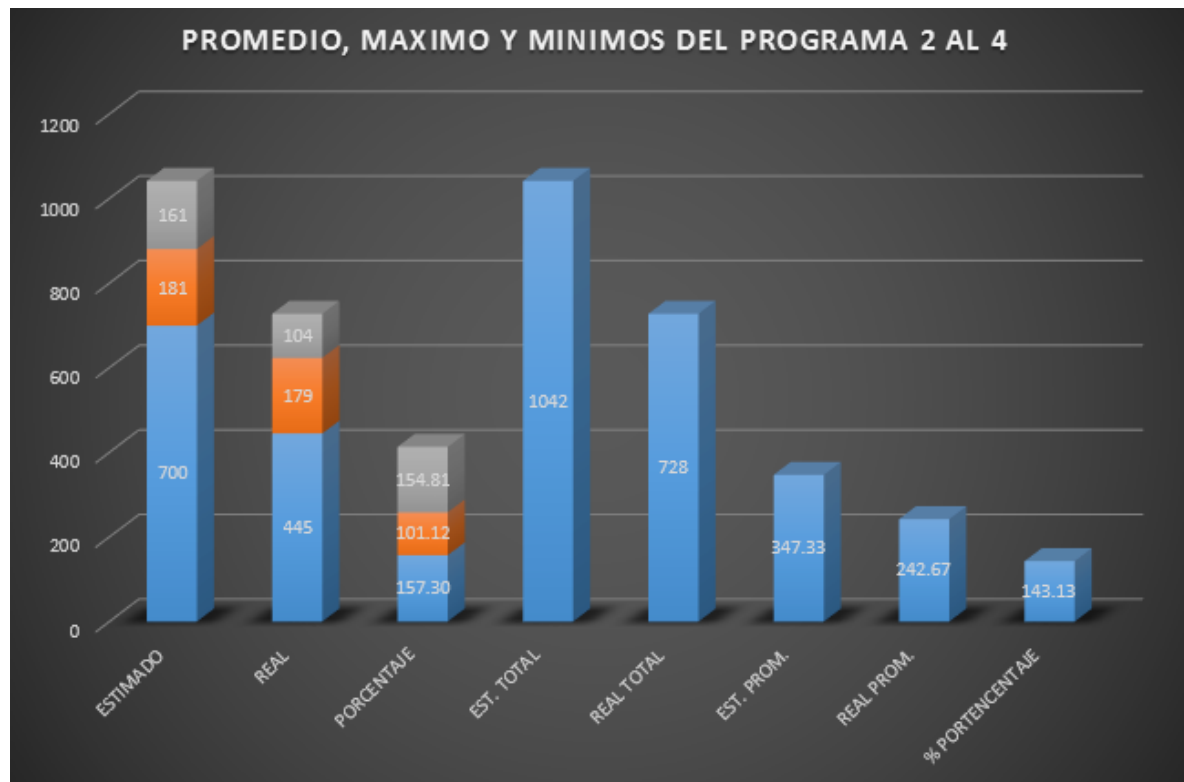


En la gráfica anterior muestra que tan certero fui e la hora de proyectar cuantas líneas necesitaba en cada programa, como se puede observar solo en el programa 3 pude acercarme a lo real ya que en el 2 y en el 4 tuve una diferencia superior del 57.30% y 54.81%, es decir que se sobreestimo, considero que esto sucedió porque ya había realizado un programa similar al 3 y por lo tanto pude deducir más certeramente cuantas líneas iba a necesitar. Creo que el planear cuantas líneas vas a utilizar en un programa depende de la experiencia del programador.

¿CUÁLES SON EL PROMEDIO, MÁXIMO Y VALORES MÍNIMOS PARA ESTOS?

EXCLUYENDO PROGRAMA 1, ¿CUÁL ES EL ERROR TOTAL ACUMULADO DE LAS ESTIMACIONES DEL TAMAÑO (ES DECIR, EL TAMAÑO TOTAL ESTIMADO / TAMAÑO REAL TOTAL)?

No. Programa	PROMEDIO, MAXIMO Y MINIMOS DEL PROGRAMA 2 AL 4							
	ESTIMADO	REAL	PORCENTAJE	EST. TOTAL	REAL TOTAL	EST. PROM.	REAL PROM.	% PORTENCENTAJE
2	700	445	157.30	1042	728	347.33	242.67	143.13
3	181	179	101.12					
4	161	104	154.81					



La gráfica muestra el total de líneas programadas esto nos arroja que tenemos un promedio de líneas reales 242.67 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 43.13% 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?

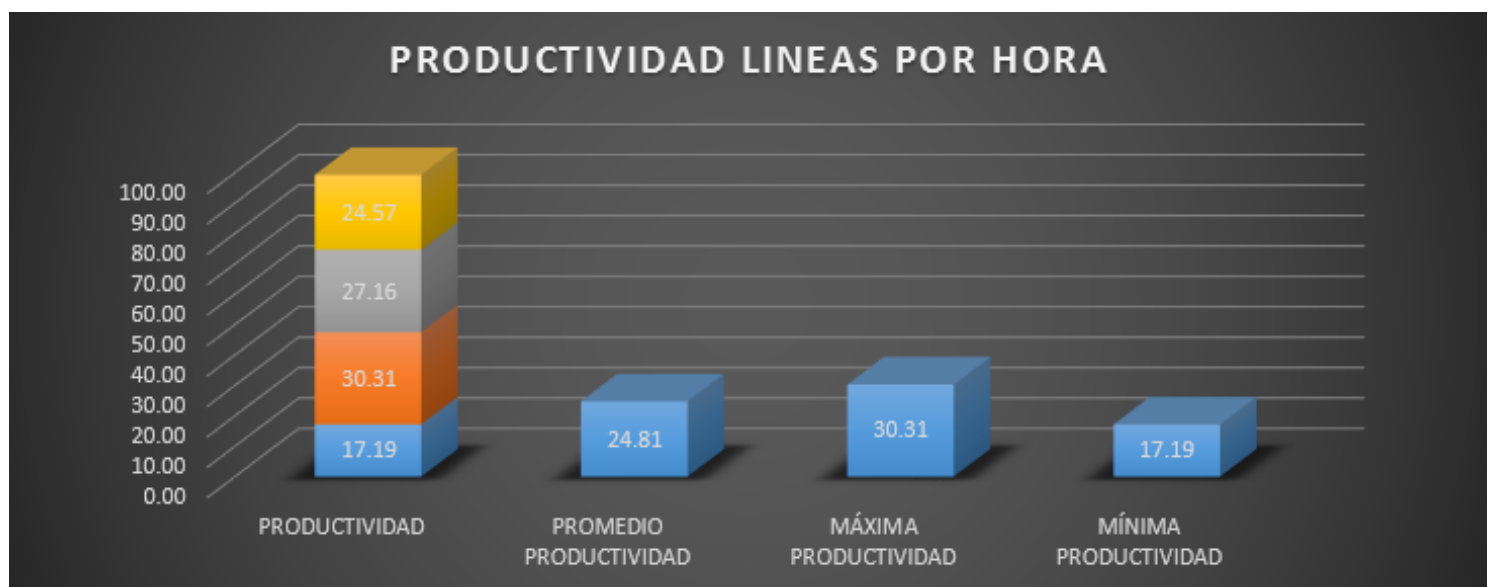
RESUMEN DEL TIEMPO EN LOS 4 PROGRAMAS				
No. Programa	MINUTOS	MEDIA	MÁXIMO	MÍNIMO
1	335	438.75	881	254
2	881			
3	285			
4	254			



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?

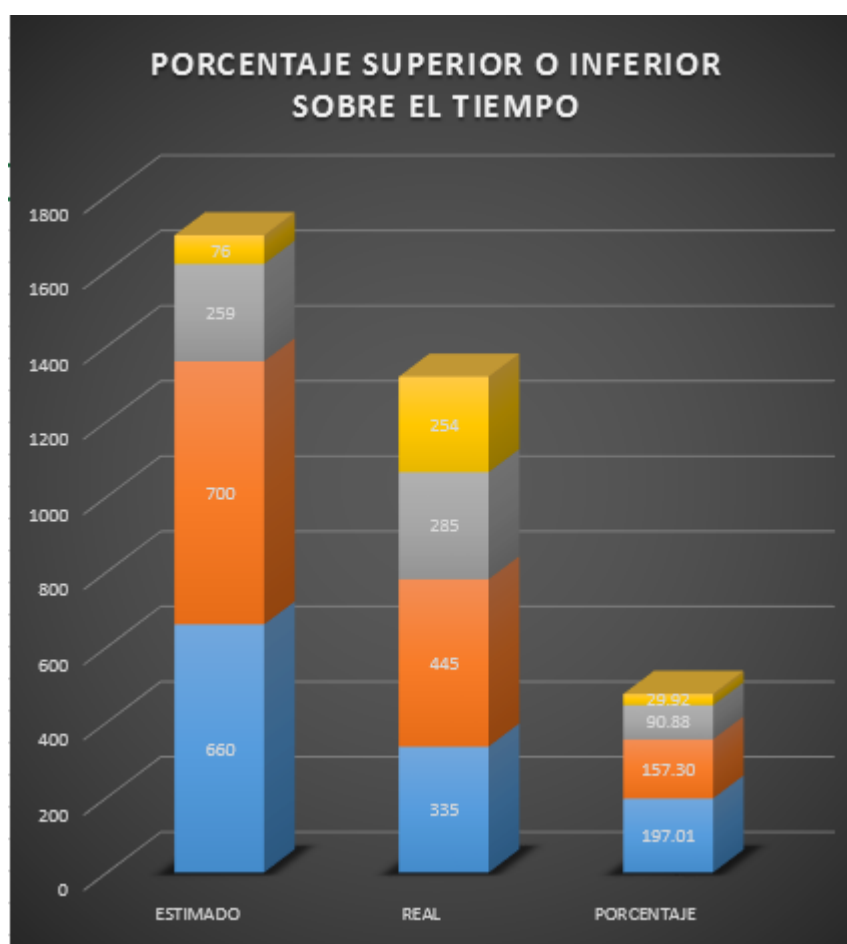
PRODUCTIVIDAD LINEAS POR HORA						
No. Programa	MINUTOS	LOC	PRODUCTIVIDAD	PROMEDIO PRODUCTIVIDAD	MÁXIMA PRODUCTIVIDAD	MÍNIMA PRODUCTIVIDAD
1	335	96	17.19	24.81	30.31	17.19
2	881	445	30.31			
3	285	129	27.16			
4	254	104	24.57			



En la gráfica anterior se muestra la productividad que obtuve en los cuatro programas donde en el programa dos 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	660	335	197.01
2	700	445	157.30
3	259	285	90.88
4	76	254	29.92



En la gráfica anterior se muestra que porcentaje de estimación obtuvo en los cuatro programas, como se puede observar en el programa uno fue donde se sobreestimo más con un 97.01% 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)?

PROMEDIO MINIMO Y MAXIMO DEL TIEMPO EMPLEADO EN LOS PROGRAMAS								
No. Programa	ESTIMADO	REAL	PORCENTAJE	EST. TOTAL	REAL TOTAL	EST. PROM.	REAL PROM.	% PORTENCENTAJE
1	660	335	197.01	1695.00	1319.00	423.75	329.75	128.51
2	700	445	157.30					
3	259	285	90.88					
4	76	254	29.92					



La gráfica muestra el total de tiempo que tarde en programar todos los programas esto nos arroja que tengo un promedio de tiempo de 329.75 minutos (5.4 horas) por programa, el programa donde menos tiempo tarde en programar fue en el programa cuatro esto creo porque código que se es muy similar en todos los métodos empleados, en el programa dos fue donde tarde más en programar esto lo atribuyo a que es el que tiene más líneas y que no contaba con base. Obtuve un error total de 28.51% sobrestimando el tiempo que estimado contra el real.

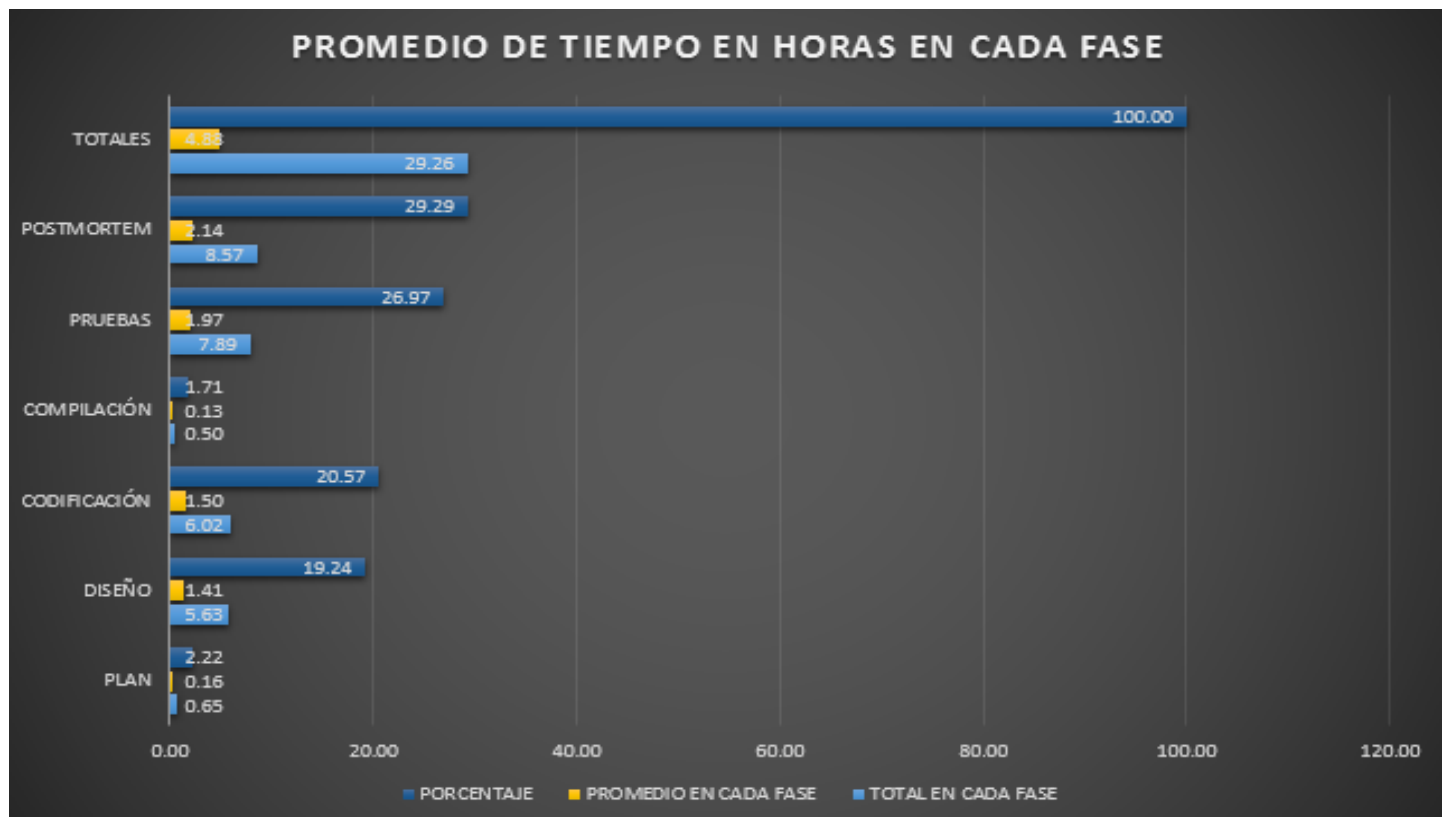
ANÁLISIS DE LA DISTRIBUCIÓN DEL TIEMPO.

¿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	0.65	0.16	2.22
DISEÑO	5.63	1.41	19.24
CODIFICACIÓN	6.02	1.50	20.57
COMPILACIÓN	0.50	0.13	1.71
PRUEBAS	7.89	1.97	26.97
POSTMORTEM	8.57	2.14	29.29
TOTALES	29.26	4.88	100.00
DISEÑO Y CÓDIGO	11.65	N/A	39.82
COMPILACIÓN Y PRUEBA	8.39	N/A	28.67



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 tuve errores.

En las fases de diseño y de código se tiene un 39% 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 28.67% en las fase de compilación y pruebas, en esta fases compruebas si tu diseño y tu código están correctos ya que en esta fases 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 19.24%, entonces se multiplica y las 1000 son el 100% del proceso, se realiza la siguiente operación $(19.24 \times 1000) / 100$ que da un resultado de 192.4 horas, estas son las horas que posiblemente se dedique a la etapa de diseño.

¿EN LA CODIFICACIÓN?

Se aplica el mismo procedimiento se multiplica $(20.57 \times 1000) / 100$ que da un resultado de 205.7 horas que es el tiempo posible que se dedicara a la fase de codificación.

¿EN PRUEBAS?

En las pruebas se multiplica $(26.97 \times 1000) / 100$ que da un resultado de 269.7 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 24.81 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 1000×24.81 y el resultado es el siguiente 24810 líneas en todo el proceso.

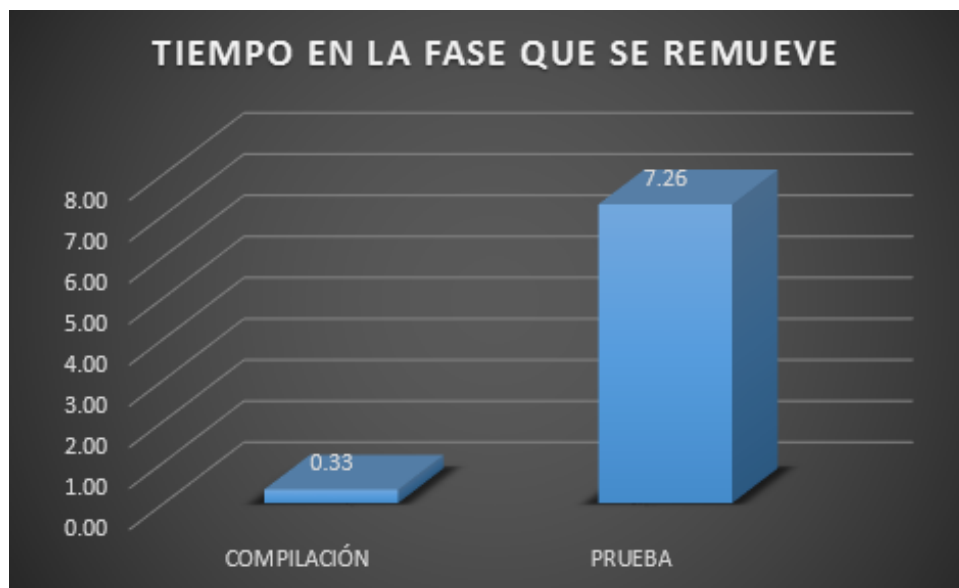
¿CUÁNTOS DEFECTOS ES PROBABLE QUE ENCUENTRE EN PRUEBAS?

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 INYECTA	TIEMPO	FASE QUE REMUEVE	TIEMPO
DISEÑO	5.15	COMPILACIÓN	0.33
CODIFICACIÓN	2.45	PRUEBA	7.26



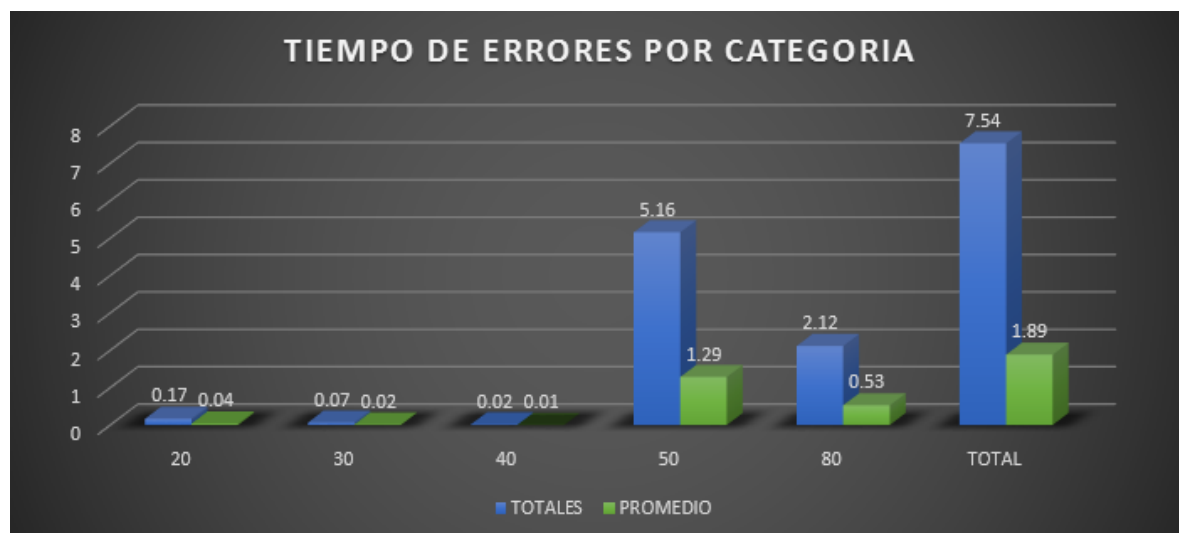
La fase en la que obtuve mayor tiempo inyectando errores es la de diseño con un total de 5.15 horas esto lo atribuyo a que estaba acostumbrado a brincarme esta fase e ir directo a la codificación, creo que este análisis me servirá para hacer un mejor diseño para tener menos errores a la hora de codificar.

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						
PROGRAMA	CATEGORIAS					TOTAL
	20	30	40	50	80	
1	0.08	0.03	0.01	0.1	2.11	2.33
2	0.08	0.01	0.01	5.06	0	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	0.02	5.16	2.12	7.54
PROMEDIO	0.04	0.02	0.01	1.29	0.53	1.89



La categoría de interfaz que es el error número 50 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	12	52.17
2	7	30.43
3	3	13.04
4	1	4.35
TOTAL	23	100.00
PROMEDIO	5.75	



En el programa uno es donde obtuve más errores con 12 en total y el cuatro donde menos obtuve errores con un total de 1 error, esto es por en el programa uno no tuve cuidado a la hora de realizar el proceso y además por falta de experiencia, conforme fue pasando cada programa obtuve menos errores, en promedio se generaron 5.75 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?

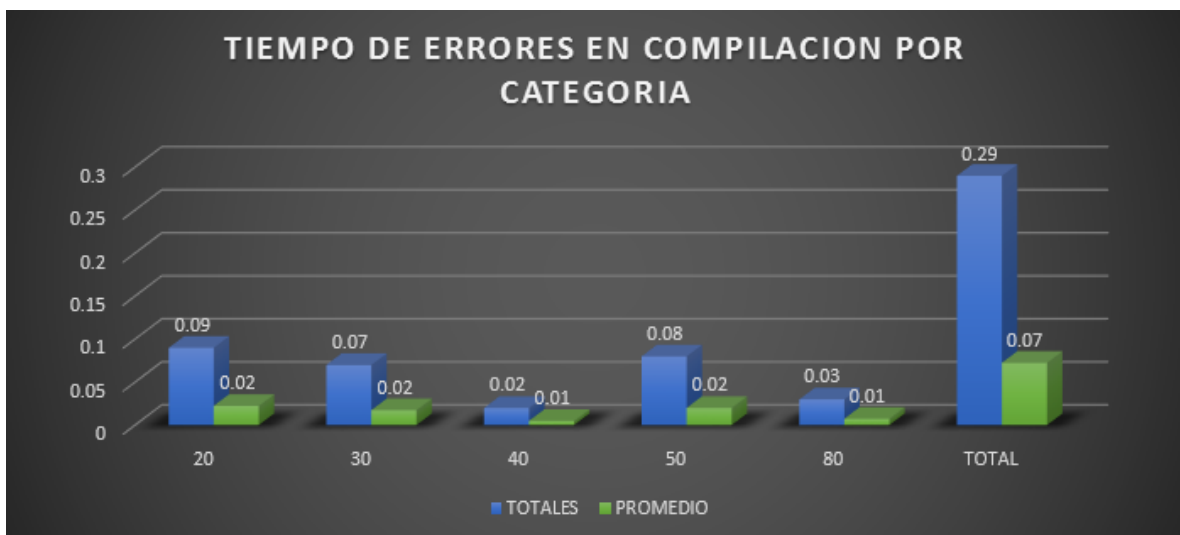
DENSIDAD DE DEFECTOS POR PROGRAMA		
No. Programa	COMPILACION	PRUEBAS
1	10.00	2.00
2	2.00	5.00
3	3.00	0.00
4	0.00	1.00
TOTAL	15.00	8.00
PROMEDIO	3.75	2.00



En la gráfica anterior se muestran los máximos, mínimos y el promedio de la densidad de defectos encontrados en las fases de compilación y prueba, en el programa uno fue donde se encuentra la mayor densidad de defectos en la fase de compilación, considero que esto paso porque no realice un buen diseño, en el programa dos 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 MAS TIEMPO EN LA FASE COMPILACIÓN?

TIEMPO DE ERRORES EN COMPILACION POR CATEGORIA						
PROGRAMA	CATEGORIAS					TOTAL
	20	30	40	50	80	
1	0.08	0.03	0.01	0.08	0.03	0.23
2	0	0.01	0.01	0	0	0.02
3	0.01	0.03	0	0	0	0.04
4	0	0	0	0	0	0
TOTALES	0.09	0.07	0.02	0.08	0.03	0.29
PROMEDIO	0.02	0.02	0.01	0.02	0.01	0.07



El defecto de tipo 20 que corresponde a sintaxis es el que más ha durado en la fase de compilación con .09 horas, 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						
PROGRAMA	CATEGORIAS					TOTAL
	20	30	40	50	80	
1	0.00	0.00	0.00	0.01	2.08	2.09
2	0.08	0.00	0.00	5.06	0.00	5.14
3	0.00	0.00	0.00	0.00	0.00	0
4	0.00	0.00	0.00	0.00	0.01	0.01
TOTALES	0.08	0	0	5.07	2.09	7.24
PROMEDIO	0.02	0.00	0.00	1.27	0.52	1.81



El defecto de tipo 50 que corresponde a la interfaz es el que más ha durado en la fase de compilación con 5.07 horas, considero que esto sucedió a que desde un inicio no se diseñó bien el programa y esto causo que no se cumplieran con los requerimientos del programa.

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

PROGRAMA	DEFECTOS POR FASE Y TIPO					TOTAL
	20	30	40	50	80	
PLAN	0	0	0	0	0	0
DISEÑO	1	1	1	2	1	6
CODIFICACIÓN	7	4	1	3	2	17
COMPILACIÓN	0	0	0	0	0	0
PRUEBAS	0	0	0	0	0	0
POSTMORTEM	0	0	0	0	0	0



La fase en que más tuve errores fue en la de codificación, el tipo de errores que se inyectaron en dicha fase son los siguientes: 20, 30, 40, 50, 80 que corresponden a sintaxis, paquetes, asignación, interfaz 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 eliminar estos errores.

PSP0 Project Plan Summary - Program 1A

Student	11050060-JORGE LUIS VALLIN M	Date	#####
Program	Correlation	Program#	1A
Instructor	J JESUS MINERO GURDADO	Language	JAVA

	Plan	Actual	To Date	To Date%
Time in Phase (min.)				
Planning		7	7	2.1
Design		57	57	17.0
Code		77	77	23.0
Compile		20	20	6.0
Test		131	131	39.1
Postmortem		43	43	12.8
Total	660.0	335	335	100.0

Defects Injected

Planning	0	0	0.0
Design	0	0	0.0
Code	12	12	100.0
Compile	0	0	0.0
Test	0	0	0.0
Total Development	12	12	100.0

Defects Removed

Planning	0	0	0.0
Design	0	0	0.0
Code	0	0	0.0
Compile	10	10	83.3
Test	2	2	16.7
Total Development	12	12	100.0
After Development	0	0	

PSP0.1 Project Plan Summary - Program 2A

Student	<u>11050060-JORGE LUIS VALLIN M</u>	Date	<u>#####</u>
Program	<u>LOC Counter</u>	Program#	<u>2A</u>
Instructor	<u>J JESUS MINERO GURDADO</u>	Language	<u>JAVA</u>

	Plan	Actual	To Date	To Date%
Program Size (LOC)				
Base(B)		<u>0</u>		
Deleted(D)		<u>0</u>		
Modified(M)		<u>0</u>		
Added(A)		<u>445</u>		
Reused(R)		<u>0</u>	<u>0</u>	
Total N&C (N)	<u>700</u>	<u>445</u>	<u>445</u>	
Total LOC(T)		<u>445</u>	<u>445</u>	
Total New Reused		<u>445</u>	<u>445</u>	

Time in Phase (min.)				
Planning	<u>6.96</u>	<u>9</u>	<u>16</u>	<u>1.3</u>
Design	<u>56.99</u>	<u>118</u>	<u>175</u>	<u>14.4</u>
Code	<u>76.98</u>	<u>126</u>	<u>203</u>	<u>16.6</u>
Compile	<u>19.99</u>	<u>3</u>	<u>23</u>	<u>1.9</u>
Test	<u>131.06</u>	<u>311</u>	<u>442</u>	<u>36.4</u>
Postmortem	<u>43.02</u>	<u>314</u>	<u>357</u>	<u>29.4</u>
Total	<u>335</u>	<u>881</u>	<u>1216</u>	<u>100.0</u>

Defects Injected				
Planning		<u>0</u>	<u>0</u>	<u>0.0</u>
Design		<u>5</u>	<u>5</u>	<u>26.3</u>
Code		<u>2</u>	<u>14</u>	<u>73.7</u>
Compile		<u>0</u>	<u>0</u>	<u>0.0</u>
Test		<u>0</u>	<u>0</u>	<u>0.0</u>
Total Development		<u>7</u>	<u>19</u>	<u>100.0</u>

Defects Removed				
Planning		<u>0</u>	<u>0</u>	<u>0.0</u>
Design		<u>0</u>	<u>0</u>	<u>0.0</u>
Code		<u>0</u>	<u>0</u>	<u>0.0</u>
Compile		<u>2</u>	<u>12</u>	<u>63.2</u>
Test		<u>5</u>	<u>7</u>	<u>36.8</u>
Total Development		<u>7</u>	<u>19</u>	<u>100.0</u>
After Development		<u>0</u>	<u>0</u>	

PSP1 Project Plan Summary - Program 3A

Student	11050060-JORGE LUIS VALLIN M	Date	#####
Program	Linear Regression	Program#	3A
Instructor	J JESUS MINERO GURDADO	Language	JAVA

	Plan	Actual	To Date	To Date%
Summary				
LOC/Hour	30.29501	27.14361	22.93732	

Program Size (LOC)

Base(B)	62	62		
Deleted(D)	0	0		
Modified(M)	12	12		
Added(A)	119	117		
Reused(R)	0	0	0	
Total N&C (N)	131	129	574	
Total LOC(T)	181	179	624	
Total New Reused	0	0	445	
Total Object LOC(E)	206	129	129	

Time in Phase (min.)

Planning	3	10	27	1.8
Design	37	74	249	16.6
Code	43	74	277	18.4
Compile	5	5	28	1.9
Test	94	27	469	31.2
Postmortem	76	94	451	30.1
Total	259	285	1501	100.0

Defects Injected

Planning	0	0	0.0
Design	0	5	22.7
Code	3	17	77.3
Compile	0	0	0.0
Test	0	0	0.0
Total Development	3	22	100.0

Defects Removed

Planning	0	0	0.0
Design	0	0	0.0
Code	0	0	0.0
Compile	3	15	68.2
Test	0	7	31.8
Total Development	3	22	100.0
After Development	0	0	

PSP1.1 Project Plan Summary - Program 4A

Student	11050060-JORGE LUIS VALLIN M	Date	#####
Program	Numerical Integration	Program#	4A
Instructor	J JESUS MINERO GURDADO	Language	JAVA

	Plan	Actual	To Date	To Date%
Summary				
LOC/Hour	29.5	24.6	23.2	
Planned Time	76		1330	
Actual Time		254	1755	
CPI (Planned/Actual Time)			0.757835	
% Reused	0	0	0	
% New Reused	0	0	0	

Program Size (LOC)

Base(B)	124	124	
Deleted(D)	0	0	
Modified(M)	0	0	
Added(A)	37	104	
Reused(R)	0	0	0
Total N&C (N)	37	104	678
Total LOC(T)	161	228	852
Total New Reused	0	0	445
Total Object LOC(E)	54	104	233

Time in Phase (min.)

Planning	1	12	39	2.2
Design	13	88	338	19.2
Code	14	84	361	20.6
Compile	1	2	30	1.7
Test	24	4	473	27.0
Postmortem	23	63	514	29.3
Total	76	254	1755	100.0

Defects Injected

Planning	0	0	0.0
Design	1	6	26.1
Code	0	17	73.9
Compile	0	0	0.0
Test	0	0	0.0
Total Development	1	23	100.0

Defects Removed

Planning	0	0	0.0
Design	0	0	0.0
Code	0	0	0.0
Compile	0	15	65.2
Test	1	8	34.8
Total Development	1	23	100.0
After Development	?????	?????	

TimeLog

Project	Phase	Date	Start	Int.	Stop	Delta	Comments
1	PLAN	01/27/15	12:26:00		12:33:00	7.0	
1	DLD	01/27/15	12:33:00		13:30:00	57.0	De 13:30 a 13:40 llamada
1	CODE	01/27/15	13:40:00		14:57:00	77.0	Comida de 14:57 a 15:57, Atención a alumnos, llamada, mensaje
1	COMPILE	01/27/15	15:57:00		16:17:00	20.0	
1	TEST	01/27/15	16:17:00		18:28:00	131.0	Se encontraron varios errores, descanso de 18:28 a 19:16
1	PM	01/27/15	19:16:00	3.0	20:02:00	43.0	llamada
2	PLAN	02/11/15	14:04:29		14:13:52	9.4	
2	DLD	02/11/15	14:13:52		16:11:52	118.0	Descanso del día 11 a las 16:11 al 12 a las 15:12
2	CODE	02/12/15	15:12:44		15:32:18	19.6	Descanso del día 12 a las 15:32 al 13 a las 13:25
2	CODE	02/13/15	13:25:35	30.0	15:41:31	105.9	Comida
2	COMPILE	02/13/15	15:55:30	2.0	16:00:36	3.1	
2	TEST	02/13/15	16:00:58	10.0	16:20:58	10.0	
2	PM	02/13/15	16:21:14	20.0	18:35:17	114.1	Descanso
2	TEST	02/16/15	23:12:37	537.0	13:10:47	301.2	Descansos
2	PM	02/20/15	18:12:33	100.0	23:12:41	200.1	Descanso
3	PLAN	03/10/15	13:54:58		14:05:19	10.4	
3	DLD	03/10/15	14:55:50		16:10:04	74.2	Descanso de 16:10 al 16:57
3	CODE	03/10/15	16:57:06	39.0	18:50:25	74.3	Descanso
3	COMPILE	03/10/15	18:50:26		18:55:34	5.1	
3	TEST	03/10/15	18:55:36		19:22:37	27.0	
3	PM	03/10/15	19:22:45	32.0	21:28:51	94.1	
4	PLAN	03/24/15	13:36:59		13:49:09	12.2	
4	DLD	03/24/15	13:50:00		15:18:29	88.5	
4	CODE	03/24/15	16:05:17	20.0	17:49:28	84.2	Descanso
4	COMPILE	03/24/15	17:49:32		17:51:18	1.8	
4	TEST	03/24/15	17:51:24		17:55:32	4.1	
4	PM	03/24/15	17:55:36		18:58:47	63.2	

DefectLog

Project	Date	Num	Type	Injected	Removed	FixTime	Fix Ref.	Description	
1	27/01/2015	1	30	CODE	COMPILE	1.0		Falto importar la librería scanner.	
1	27/01/2015	2	20	CODE	COMPILE	2.0		No se creó el objeto para crear la lista y de esta forma poder almacenar los datos	
1	27/01/2015	3	20	CODE	COMPILE	1.0	2	Se escribió mal la palabra ListaEnalzada que es la clase de donde se mandan llamar los métodos.	
1	27/01/2015	4	40	CODE	COMPILE	1.0		A la hora de leer el valor, se lee como entero y la variable estaba esperando una valor doublé.	
1	27/01/2015	5	50	CODE	COMPILE	3.0		A la hora de mandar llamar el método obtener y guardarlo en una variable, faltó poner el tipo de valor que está almacenado.	
1	27/01/2015	6	30	CODE	COMPILE	1.0		Faltó importar la librería DecimalFormat para poder dar formato a los resultados.	
1	27/01/2015	7	50	CODE	COMPILE	2.0	6	No se construyó el objeto de DecimalFormat para poder hacer uso de esa función.	
1	27/01/2015	8	20	CODE	COMPILE	1.0		Se colocó mal un paréntesis en la línea para imprim	
1	27/01/2015	9	80	CODE	COMPILE	2.0		Faltó insertar la función para poder elevar al cuadrado. Si se diseñó pero no se colocó a la hora de programar.	
1	27/01/2015	10	20	CODE	COMPILE	1.0		Se colocó mal una llave en el primer for de la clase Main, al revisar visualmente el código se detectó este error.	
1	27/01/2015	11	80	CODE	TEST	125.0		El método para obtener los valores guardados en la lista tenía varios errores, la cabeza del nodo no estaba apuntando al inicio de la lista y la condición	
1	27/01/2015	12	50	CODE	TEST	1.0		Faltó dar formato al resultado de la desviación estándar. Si se diseñó pero no se respetó el diseño.	
2	13/02/2015	13	30	DLD	COMPILE	1.0		Falto importar la librería java.io.*.	
2	13/02/2015	14	40	DLD	COMPILE	1.0		No se inicializo la variable archivo	
2	13/02/2015	15	20	DLD	TEST	2.0		Falto crear el objeto de la clase FileReader para leer el segundo archivo	
2	13/02/2015	16	50	DLD	TEST	4.0		Falto colocar las variables y mandar imprimir a pantalla el total de líneas de ambos programas	
2	13/02/2015	17	20	CODE	TEST	1.0	16	Por error se repitio una de las variables para sacar el total de lineas	
2	13/02/2015	18	20	CODE	TEST	2.0		Se coloco mal una llava del primer for y no cumplia con la condicion	
2	16/02/2015	19	50	DLD	TEST	300.0		No se cumplia con los requerimientos del progrma	
3	10/03/2015	20	30	CODE	COMPILE	1.0		Faltó importar la librería DecimalFormat para poder dar formato a los resultados.	
3	10/03/2015	21	30	CODE	COMPILE	1.0		Falto importar la librería scanner.	
3	10/03/2015	22	20	CODE	COMPILE	1.0		No se creó el objeto para crear la lista	
4	24/03/2015	23	80	DLD	TEST	1.0		Se coloco mal un parentesis en la formula de desviasion estandar	