



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

CARRERA:

INGENIERIA EN SISTEMAS COMPUTACIONALES

ASIGNATURA:

PROCESO PERSONAL DE DESARROLLO DE SOFTWARE (PSP)

NOMBRE DE LA PRÁCTICA:

APLICACIÓN DEL PROCESO PSP1

NÚMERO DE LA PRÁCTICA:

03

INTEGRANTES DEL EQUIPO:

GONZALO MARTÍNEZ SILVERIO

DOCENTE:

DRA TANIA TURRUBIATES LOPEZ

SEMESTRE

7

GRUPO:

1A





INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

INTRODUCCIÓN:

PSP1 es un enfoque estructurado para el desarrollo de software que se centra en la precisión de las estimaciones de tamaño y en la mejora continua de los procesos de desarrollo. Ayuda a los ingenieros de software a planificar y gestionar proyectos de manera más efectiva, lo que a su vez conduce a una mayor calidad y productividad en la industria del desarrollo de software.

OBJETIVO:

El objetivo principal de PSP1 es establecer un procedimiento ordenado y repetible para desarrollar estimaciones precisas del tamaño del software. Esto es esencial para una gestión efectiva del desarrollo de software y para mejorar la calidad y la productividad en la ingeniería del software.

COMPETENCIA A DESARROLLAR:

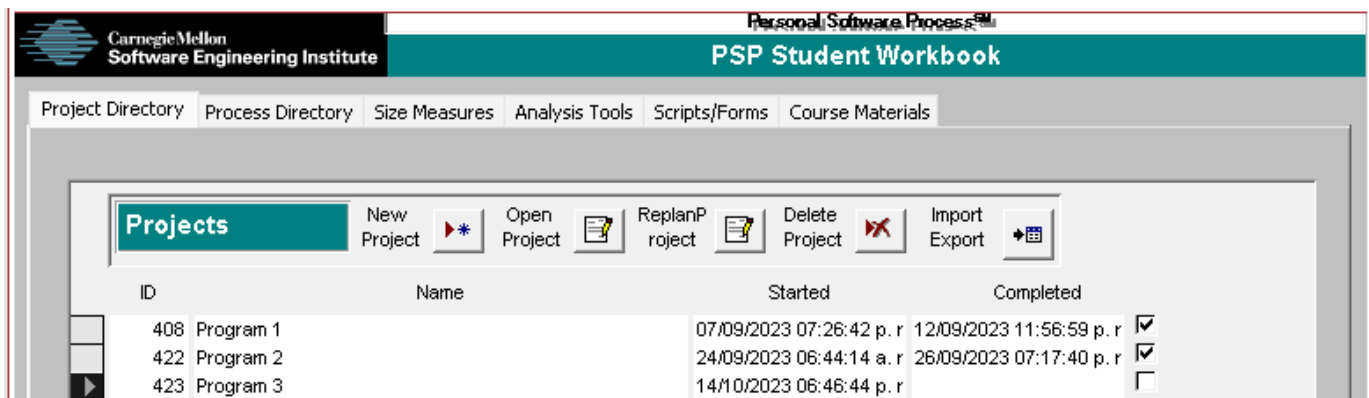
Que el alumno conozca su ritmo de trabajo y pueda hacer una evaluación del tiempo que tarda y con respecto a ello conozca su ritmo de trabajo en cada etapa.

MATERIAL Y EQUIPO (REQUERIMIENTOS):

- | |
|---------------------------------|
| • Laptop personal. |
| • Microsoft Office Word. |
| • JDK 8u202. |
| • IDE NetBeans 8.2. |
| • Material de prácticas de PSP. |

DESARROLLO:

Primero registramos el inicio de la etapa de planeación en la herramienta **PSP Student Workbook.mde**:





ETAPA DE PLANEACION:

Al igual que en el paso anterior se debe registrar en el time log el inicio de la etapa de planeación:

ID	PID	Phase	Start	Int.
34	423	PLAN	14/10/2023 06:49:58 p.m.	0

1.-Requerimientos del programa:

Escribir un programa para calcular los parámetros de regresión lineal β_0 y β_1 y los coeficientes de correlación $r_{x,y}$ y r^2 para un conjunto de n pares de datos.

- Dada una estimación, X_k calcula una predicción mejorada, y_k donde $y_k = \beta_0 + \beta_1 X_k$
- Mejorar la lista enlazada desarrollada en el programa 1 para almacenar los n conjuntos de datos, donde cada registro contiene dos números reales
- La Tabla 1 contiene datos históricos estimados y reales para 10 programas. Para el programa 11, el desarrollador ha estimado un tamaño de proxy de 386 LOC.

Pruebe minuciosamente el programa. Como mínimo, ejecute los siguientes cuatro casos de prueba.

- Prueba 1: Calcule los parámetros de regresión y los coeficientes de correlación entre el tamaño aproximado estimado y el tamaño real agregado y modificado en la Tabla 1. Calcule el tamaño agregado y modificado del plan dado un tamaño de proxy estimado de $X_k=386$.
- Prueba 2: Calcule los parámetros de regresión y los coeficientes de correlación entre el tamaño del proxy estimado y el tiempo de desarrollo real en la Tabla 1. Calcule el tiempo estimado dado un tamaño de proxy estimado de $X_k=386$.
- Prueba 3: Calcule los parámetros de regresión y los coeficientes de correlación entre el tamaño agregado y modificado del plan y el tamaño agregado y modificado real en



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

Tabla 1. Calcule el tamaño agregado y modificado del plan dado un tamaño de proxy estimado de $X_k=386$.

- Prueba 4: Calcule los parámetros de regresión y los coeficientes de correlación entre el tamaño agregado y modificado del plan y el tiempo de desarrollo real en la Tabla Calcule el tiempo estimado dado un tamaño de proxy estimado de $X_k=386$.

Número de programa	Tamaño aproximado estimado	Tamaño agregado y modificado del plan	Tamaño agregado y modificado real	Horas de desarrollo real
1	130	163	186	15.0
2	650	765	699	69.9
3	99	141	132	6.5
4	150	166	272	22.4
5	128	137	291	28.4
6	302	355	331	65.9
7	95	136	199	19.4
8	945	1206	1890	198.7
9	368	433	788	38.8
10	961	1130	160138.2	138.2

Tabla 1

Los resultados esperados se proporcionan en la Tabla 2.

Pruebas	β_0	β_1	$r_{x,y}$	r^2	y_k
1	-22.55	1.7279	0.9545	0.9111	644.429
2	-4.039	0.1681	0.9333	.8711	60.858
3	-23.92	1.43097	.9631	.9276	528.4294
4	-4.604	0.140164	.9480	.8988	49.4994

La regresión lineal es una forma de ajustar de manera óptima una línea a un conjunto de datos. La línea de regresión lineal es la línea donde se minimiza la distancia desde todos los puntos a esa línea. La ecuación de una recta se puede escribir como $y=\beta_0+\beta_1X$

En la Figura 1, la línea de regresión de mejor ajuste tiene parámetros de

$\beta_0= -4,0389$ y $\beta_1=0,1681$

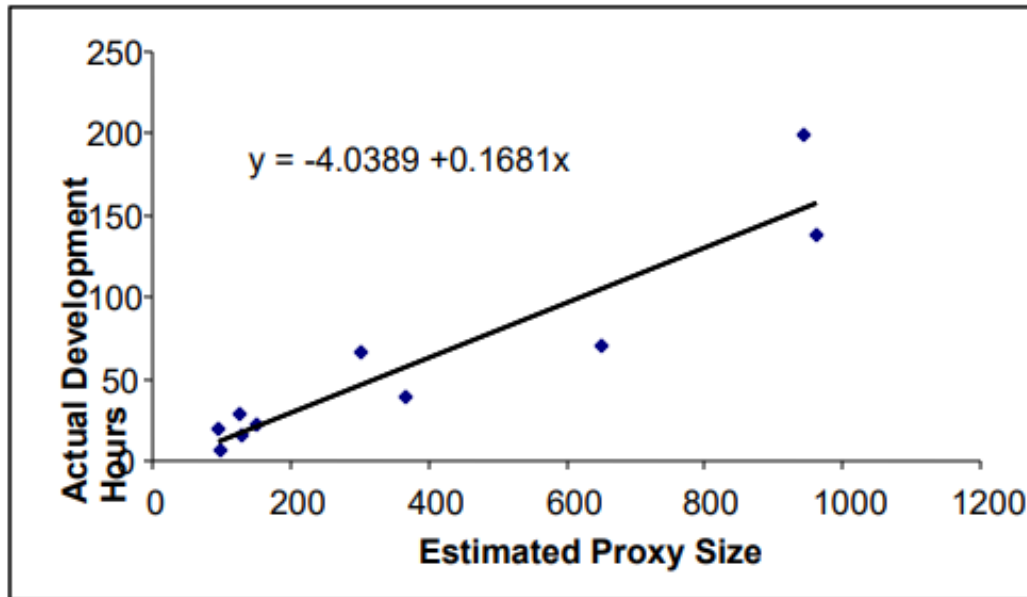


Figura 1

2.- Estimación de tiempo:

El tiempo estimado para este proyecto en todas sus etapas puesto que los requerimientos son bastante específicos y su estructura poco compleja será de 2 horas y 50 minutos.

3.-Registramos la conclusión de la etapa de planeación registrándolo en el Time Log:

The screenshot shows the 'PSP Time Recording Log' window. At the top, it says 'Carnegie Mellon Software Engineering Institute' and 'Personal Software Process'. Below this, there are fields for 'Student' (Gonzalo Martinez Silverio), 'Program' (Program 3), and 'Instructor' (TANIA TURRUBIATES LOPEZ). To the right, there are fields for 'Start Date' (14-oct.-23), 'End Date', and 'Language' (Java). The ID is 423. Below these fields is a table with the following columns: ID, PID, Phase, Start, Int., Stop Date and Time, Delta, and Comments. The table contains one entry with ID 34, PID 423, Phase PLAN, Start 14/10/2023 06:49:58 p. r, Int. 0, Stop Date and Time 14/10/2023 07:20:10 p. r, Delta 30.2, and an empty Comments field.

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
34	423	PLAN	14/10/2023 06:49:58 p. r	0	14/10/2023 07:20:10 p. r	30.2	

ETAPA DE DISEÑO:

1.-Se registra el inicio de la etapa de diseño en el Time Log:



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute PSP Time Record

Student: Gonzalo Martinez Silverio
Program: Program 3
Instructor: TANIA TURRUBIATES LOPEZ

ID	PID	Phase	Start	Int.
34	423	PLAN	14/10/2023 06:49:58 p.r	0
35	423	DLD	14/10/2023 07:20:11 p.r	0

2.-El diseño que se presentará será el siguiente:

El programa constara de una clase la cual implementara la regresión lineal para analizar la relación entre datos de "EstimatedProxySize" y "ActualAddedandModifiedSize." Calculará coeficientes de regresión, correlación y predicciones para múltiples conjuntos de datos. Los resultados se imprimirán en la consola.

Las clases quedarían estructuradas de la siguiente manera:

```
LinearRegression
- obtenerDatos(tipo: String, scanner: Scanner): ArrayList<Double>
- calcularYk(xEstimado: ArrayList<Double>, yReal: ArrayList<Double>, xk: ArrayList<Double>, conjunto: int): double[]
- calcularCoeficienteCorrelacion(x: ArrayList<Double>, y: ArrayList<Double>): double
- calcularMedia(lista: ArrayList<Double>): double
- calcularBeta0(x: ArrayList<Double>, y: ArrayList<Double>): double
- calcularBeta1(x: ArrayList<Double>, y: ArrayList<Double>): double
- calcularR2(x: ArrayList<Double>, y: ArrayList<Double>): double
- predecirY(beta0: double, beta1: double, xk: double): double
+ main(args: String[]): void
```

3.-Finalizamos la etapa de Diseño y la registramos en el Time Log:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute Personal Software Process PSP Time Recording Log

Student: Gonzalo Martinez Silverio Start Date: 14-oct.-23 ID: 423
Program: Program 3 End Date:
Instructor: TANIA TURRUBIATES LOPEZ Language: Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
34	423	PLAN	14/10/2023 06:49:58 p.r	0	14/10/2023 07:20:10 p.r	30.2	
35	423	DLD	14/10/2023 07:20:11 p.r	0	14/10/2023 08:17:59 p.r	57.8	

ETAPA DE CODIFICACION:

Registramos el inicio de la etapa de codificación en nuestro Time Log:



REPORTE DE PRÁCTICAS

PSP Time Recording Log

Carnegie Mellon
Software Engineering Institute

Student: Gonzalo Martinez Silverio
Program: Program 3
Instructor: TANIA TURRUBIATES LOPEZ

ID	PID	Phase	Start
34	423	PLAN	14/10/2023 06:49:58 p. r
35	423	DLD	14/10/2023 07:20:11 p. r
36	423	CODE	14/10/2023 08:18:00 p. r
37	423		

Creamos la clase principal y declaramos las variables que vamos a necesitar como las fórmulas para obtener la regresión lineal. Calculamos los coeficientes de regresión β_0 y β_1 , el coeficiente de correlación r_{xy} , el coeficiente de determinación r^2 y realiza predicciones y_k para varios conjuntos de datos.

A continuación, se explican las principales funciones:

obtenerDatos(Scanner scanner, String tipo): Esta función recopila datos de entrada del usuario a través de la consola. Toma como argumento un objeto Scanner para la entrada estándar y una cadena tipo que se utiliza para indicar qué tipo de datos se está recopilando. Los datos ingresados por el usuario se almacenan en una lista enlazada (*LinkedList*) de números de punto flotante.

calcularMedia(LinkedList<Double> lista): Esta función calcula la media aritmética de una lista de números de punto flotante.

calcularCoeficienteCorrelacion(LinkedList<Double> x, LinkedList<Double> y): Calcula el coeficiente de correlación (r) entre dos conjuntos de datos, en este caso, entre "Tamaño de proxy estimado" (x) y "Tamaño real agregado y modificado" (y).

calcularBeta0(LinkedList<Double> x, LinkedList<Double> y) y calcularBeta1(LinkedList<Double> x, LinkedList<Double> y): Estas funciones calculan los coeficientes de la regresión lineal (β_0 y β_1) a partir de los datos de entrada.

calcularR2(LinkedList<Double> x, LinkedList<Double> y): Calcula el coeficiente de determinación (r^2), que indica cuánta variabilidad en la variable dependiente se puede explicar mediante la variable independiente.

predecirY(double beta0, double beta1, double xk): Predice el valor de la variable dependiente (*Tamaño real agregado y modificado*) (y_k) para un valor dado de la variable independiente (*Tamaño de proxy estimado*) (x_k), utilizando los coeficientes calculados.

En el método *main*, se realiza la ejecución del programa, donde se obtienen datos de entrada para cuatro conjuntos (*pruebas 1 a 4*) y se calculan los resultados de la regresión lineal, incluyendo los coeficientes β_0 y β_1 , el coeficiente de correlación r_{xy} , el coeficiente de determinación r^2 y las



predicciones y_k para cada conjunto.

```
PSP - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Start Page x RegresionLineal.java x
Source History
1 package PROGRAMA3;
2 /**
3  * INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE
4  * @AUTOR: Gonzalo Martinez Silverio
5  * No. Control: 20220029
6  * ASIGNATURA: Process Personal Software
7  * DOCENTE: Dra Tania Turrubiates Lopez
8  */
9 import java.util.*;
10
11 public class RegresionLineal {
12
13     public static LinkedList<Double> obtenerDatos(Scanner scanner, String tipo) {
14         System.out.println("Ingrese los datos de " + tipo + " (separados por espacios):");
15         String[] input = scanner.nextLine().split(" ");
16         LinkedList<Double> datos = new LinkedList<Double>();
17         for (String dato : input) {
18             datos.add(Double.parseDouble(dato));
19         }
20         return datos;
21     }
22
23     public static void calcularYk(LinkedList<Double> xEstimado,
24         LinkedList<Double> yReal, LinkedList<Double> xk, int conjunto) {
25         double beta0 = calcularBeta0(xEstimado, yReal);
26         double beta1 = calcularBeta1(xEstimado, yReal);
27         double rxy = calcularCoeficienteCorrelacion(xEstimado, yReal);
28         double r2 = calcularR2(xEstimado, yReal);
29         double yk = predecirY(beta0, beta1, xk.get(0));
30     }
31 }
```




REPORTE DE PRÁCTICAS

```
30
31     System.out.println("Resultados para el conjunto " + conjunto + ":");
32     System.out.println("B_0 = " + beta0);
33     System.out.println("B_1 = " + beta1);
34     System.out.println("r_(x,y) = " + rxy);
35     System.out.println("r^2 = " + r2);
36     System.out.println("y_k = " + yk);
37 }
38
39 // Define aquí las funciones calcularBeta0, calcularBeta1,
40 // calcularCoeficienteCorrelacion, calcularR2 y predecirY
41 public static double calcularMedia(LinkedList<Double> lista) {
42     double suma = 0.0;
43     for (double valor : lista) {
44         suma += valor;
45     }
46     return suma / lista.size();
47 }
48
49 public static double calcularCoeficienteCorrelacion(LinkedList<Double> x, LinkedList<Double> y) {
50     double mediaX = calcularMedia(x);
51     double mediaY = calcularMedia(y);
52
53     double numerador = 0.0;
54     double denominadorX = 0.0;
55     double denominadorY = 0.0;
56
57     for (int i = 0; i < x.size(); i++) {
58         numerador += (x.get(i) - mediaX) * (y.get(i) - mediaY);
59         denominadorX += Math.pow(x.get(i) - mediaX, 2);
60         denominadorY += Math.pow(y.get(i) - mediaY, 2);
61     }
62     return numerador / (Math.sqrt(denominadorX) * Math.sqrt(denominadorY));
63 }
64
65 public static double calcularBeta0(LinkedList<Double> x, LinkedList<Double> y) {
66     double mediaX = calcularMedia(x);
67     double mediaY = calcularMedia(y);
68     double rxy = calcularCoeficienteCorrelacion(x, y);
69     double beta1 = calcularBeta1(x, y);
70     return mediaY - beta1 * mediaX;
71 }
72
73 public static double calcularBeta1(LinkedList<Double> x, LinkedList<Double> y) {
74     double mediaX = calcularMedia(x);
75     double mediaY = calcularMedia(y);
76     double numerador = 0.0;
77     double denominador = 0.0;
78
79     for (int i = 0; i < x.size(); i++) {
80         numerador += (x.get(i) - mediaX) * (y.get(i) - mediaY);
81         denominador += Math.pow(x.get(i) - mediaX, 2);
82     }
```



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

```
82 |         return numerador / denominador;
83 |     }
84 |
85 |     public static double calcularR2(LinkedList<Double> x, LinkedList<Double> y) {
86 |         double rxy = calcularCoeficienteCorrelacion(x, y);
87 |         return rxy * rxy;
88 |     }
89 |
90 |     public static double predecirY(double beta0, double beta1, double xk) {
91 |         return beta0 + beta1 * xk;
92 |     }
93 |
94 |     public static void main(String[] args) {
95 |         Scanner scanner = new Scanner(System.in);
96 |         // Datos de entrada
97 |         // Prueba 1
98 |         System.out.println(" * INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE\n" + " * @AUTOR: Gonzalo");
99 |         System.out.println("Conjunto 1:");
100 |         LinkedList<Double> xEstimado = obtenerDatos(scanner, "Tamaño de proxy estimado");
101 |         LinkedList<Double> yReal = obtenerDatos(scanner, "Tamaño real agregado y modificado");
102 |         LinkedList<Double> xk = obtenerDatos(scanner, "Proxy (Para las cuatro pruebas)\n");
103 |         calcularYk(xEstimado, yReal, xk, 1);
104 |
105 |         // Prueba 2
106 |         System.out.println("\nConjunto 2:");
107 |         LinkedList<Double> xEstimado2 = obtenerDatos(scanner, "Tamaño de proxy estimado");
108 |         LinkedList<Double> yReal2 = obtenerDatos(scanner, "Tamaño real agregado y modificado");
109 |         calcularYk(xEstimado2, yReal2, xk, 2);
110 |
111 |         // Prueba 3
112 |         System.out.println("\nConjunto 3:\n");
113 |         LinkedList<Double> xEstimado3 = obtenerDatos(scanner, "Tamaño de proxy estimado");
114 |         LinkedList<Double> yReal3 = obtenerDatos(scanner, "Tamaño real agregado y modificado");
115 |         calcularYk(xEstimado3, yReal3, xk, 3);
116 |
117 |         // Prueba 4
118 |         System.out.println("\nConjunto 4:\n");
119 |         LinkedList<Double> xEstimado4 = obtenerDatos(scanner, "Tamaño de proxy estimado");
120 |         LinkedList<Double> yReal4 = obtenerDatos(scanner, "Tamaño real agregado y modificado");
121 |         calcularYk(xEstimado4, yReal4, xk, 4);
122 |     }
123 | }
```

3.-Marcamos como finalizada la etapa de codificación en el Time Log:

PSP Time Recording Log

PSP Time Recording Log		ID 423					
Student	Gonzalo Martinez Silverio	Start Date	14-oct.-23				
Program	Program 3	End Date					
Instructor	TANIA TURRUBIATES LOPEZ	Language	Java				
ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
34	423	PLAN	14/10/2023 06:49:58 p. r	0	14/10/2023 07:12:10 p. r	22.2	Estimacion y planeacion
35	423	DLD	14/10/2023 07:12:11 p. r	0	14/10/2023 08:01:35 p. r	49.4	Diseño del diagrama UML
36	423	CODE	14/10/2023 08:01:36 p. r	734	15/10/2023 09:17:09 a. r	61.6	Coficacion / interrupcion para dormir



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

ETAPA DE COMPILACION:

BUILD SUCCESSFUL total time: 2

Se compila:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute Personal Software ProcessSM

PSP Time Recording Log

Student: Gonzalo Martinez Silverio Start Date: 14-oct.-23 ID: 423

Program: Program 3 End Date:

Instructor: TANIA TURRUBIATES LOPEZ Language: Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
34	423	PLAN	14/10/2023 06:49:58 p. r	0	14/10/2023 07:12:10 p. r	22.2	Estimacion y planeacion
35	423	DLD	14/10/2023 07:12:11 p. r	0	14/10/2023 08:01:35 p. r	49.4	Diseño del diagrama UML
36	423	CODE	14/10/2023 08:01:36 p. r	734	15/10/2023 09:17:09 a. r	61.6	Coficacion / interrupcion para dormir
37	423	COMPILE	15/10/2023 09:17:10 a. r	0	15/10/2023 09:19:10 a. r	2.0	Compilacion
38	423	UT	15/10/2023 09:19:10 p. r	0		0.0	Pruebas

Antes de finalizar esta etapa, se optó por llevar a cabo una compilación y realizar una prueba con la introducción de datos de la Tabla 1. Como no se identificó ningún error durante esta fase entonces procederé a la siguiente etapa.

ETAPA DE TESTEO:

Registramos el inicio de nuestra etapa de Testeo:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute Personal Software ProcessSM

PSP Time Recording Log

Student: Gonzalo Martinez Silverio Start Date: 14-oct.-23 ID: 423

Program: Program 3 End Date:

Instructor: TANIA TURRUBIATES LOPEZ Language: Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
34	423	PLAN	14/10/2023 06:49:58 p. r	0	14/10/2023 07:12:10 p. r	22.2	Estimacion y planeacion
35	423	DLD	14/10/2023 07:12:11 p. r	0	14/10/2023 08:01:35 p. r	49.4	Diseño del diagrama UML
36	423	CODE	14/10/2023 08:01:36 p. r	734	15/10/2023 09:17:09 a. r	61.6	Coficacion / interrupcion para dormir
37	423	COMPILE	15/10/2023 09:17:10 a. r	0	15/10/2023 09:19:10 a. r	2.0	Compilacion
38	423	UT	15/10/2023 09:19:10 p. r	0		0.0	Pruebas



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

Ahora procedemos a ingresar los datos de la Tabla 1:

```
Start Page x RegresionLineal.java x Output - PSP (run) x

run:
* INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE
* @AUTOR: Gonzalo Martinez Silverio
* No. Control: 20220029
* ASIGNATURA: Process Personal Software
* DOCENTE: Dra Tania Turrubiates Lopez

Conjunto 1:
Ingrese los datos de Tamaño de proxy estimado (separados por espacios):
```

Introducimos los datos:

```
PSP - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page x RegresionLineal.java x Output - PSP (run) x

run:
* INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE
* @AUTOR: Gonzalo Martinez Silverio
* No. Control: 20220029
* ASIGNATURA: Process Personal Software
* DOCENTE: Dra Tania Turrubiates Lopez

Conjunto 1:
Ingrese los datos de Tamaño de proxy estimado (separados por espacios):
130 650 99 150 128 302 95 945 368 961
Ingrese los datos de Tamaño real agregado y modificado (separados por espacios):
186 699 132 272 291 331 199 1890 788 1601
Ingrese los datos de Proxy (Para las cuatro pruebas) (separados por espacios):
386 386 386 386

Conjunto 2:
Ingrese los datos de Tamaño de proxy estimado (separados por espacios):
130 650 99 150 128 302 95 945 368 961
Ingrese los datos de Tamaño real agregado y modificado (separados por espacios):
15.0 69.9 6.5 22.4 28.4 65.9 19.4 198.7 38.8 138.2

Conjunto 3:
Ingrese los datos de Tamaño de proxy estimado (separados por espacios):
163 765 141 166 137 355 136 1206 433 1130
Ingrese los datos de Tamaño real agregado y modificado (separados por espacios):
186 699 132 272 291 331 199 1890 788 1601
```



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

Conjunto 4;

Ingrese los datos de Tamaño de proxy estimado (separados por espacios):

163 765 141 166 137 355 136 1206 433 1130

Ingrese los datos de Tamaño real agregado y modificado (separados por espacios):

15.0 69.9 6.5 22.4 28.4 65.9 19.4 198.7 38.8 138.2

Resultados:

Resultados para el conjunto 1:

$B_0 = -22.552532752034267$

$B_1 = 1.727932426206986$

$r_{(x,y)} = 0.9544965741046827$

$r^2 = 0.911063709977576$

$y_k = 644.4293837638623$

Resultados para el conjunto 2:

$B_0 = -4.038881574687579$

$B_1 = 0.168126649881629$

$r_{(x,y)} = 0.9333068981405511$

$r^2 = 0.871061766116737$

$y_k = 60.858005279621224$

Resultados para el conjunto 3:

$B_0 = -23.92388825291539$

$B_1 = 1.430966943551199$

$r_{(x,y)} = 0.9631140931490526$

$r^2 = 0.927588756422322$

$y_k = 528.4293519578474$

Resultados para el conjunto 4:

$B_0 = -4.603745423308979$

$B_1 = 0.14016352638883633$

$r_{(x,y)} = 0.9480329874300507$

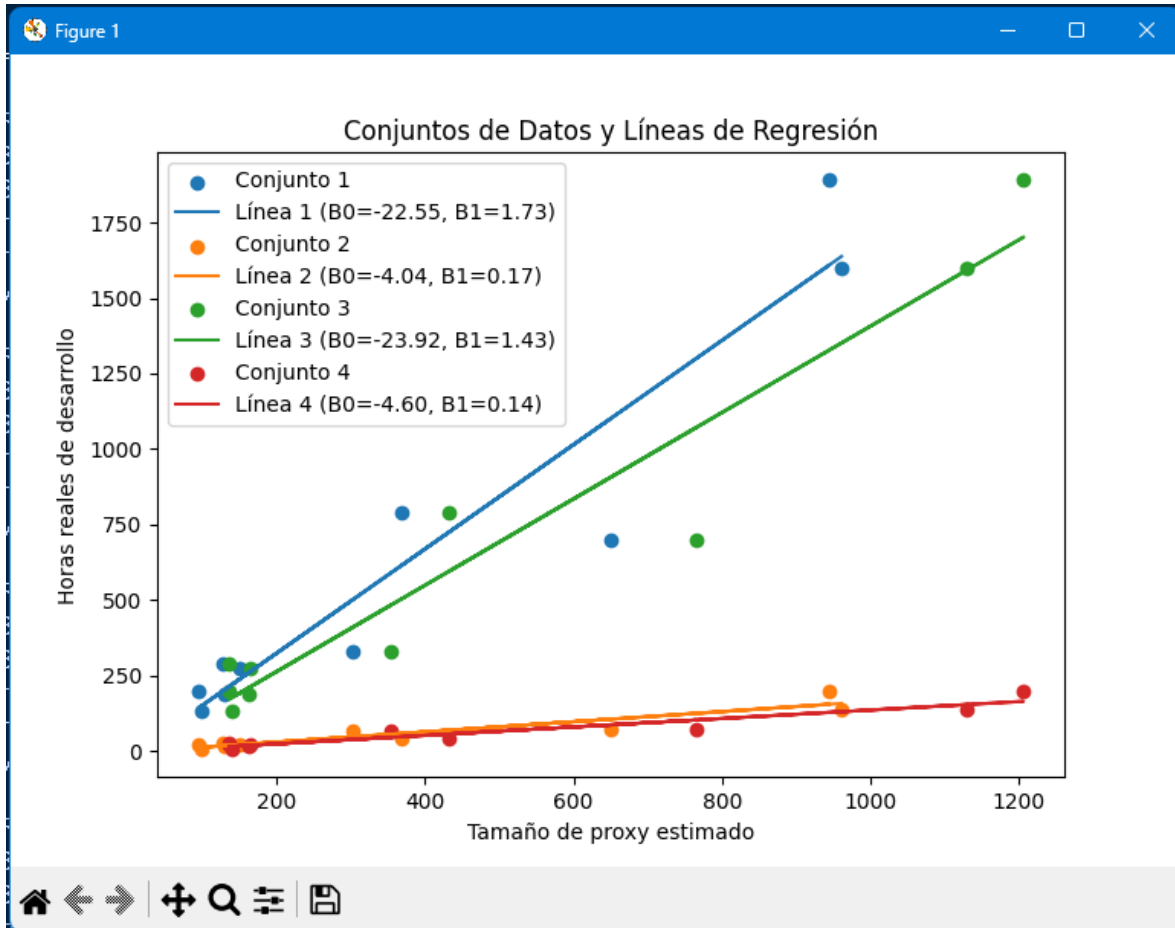
$r^2 = 0.8987665452555467$

$y_k = 49.499375762781845$

BUILD SUCCESSFUL (total time: 1 minute 37 seconds)



REPORTE DE PRÁCTICAS



Los resultados fueron verificados y en efecto son correctos y los mismo por lo tanto marcamos como finalizada la etapa de pruebas:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Student: Gonzalo Martinez Silverio

Program: Program 3

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 14-oct.-23

End Date:

Language: Java

ID: 423

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
34	423	PLAN	14/10/2023 06:49:58 p. r	0	14/10/2023 07:12:10 p. r	22.2	Estimacion y planeacion
35	423	DLD	14/10/2023 07:12:11 p. r	0	14/10/2023 08:01:35 p. r	49.4	Diseño del diagrama UML
36	423	CODE	14/10/2023 08:01:36 p. r	734	15/10/2023 09:17:09 a. r	61.6	Coficacion / interrupcion para dormir
37	423	COMPILE	15/10/2023 09:17:10 a. r	0	15/10/2023 09:19:10 a. r	2.0	Compilation
38	423	UT	15/10/2023 09:19:10 p. r	0	15/10/2023 09:29:21 p. r	10.2	Pruebas



ETAPA POSMORTEM:

Registramos el inicio de la etapa:

The screenshot shows the 'PSP Time Recording Log' window. At the top, it displays the Carnegie Mellon Software Engineering Institute logo and the title 'PSP Time Recording Log'. Below this, there are fields for Student (Gonzalo Martinez Silverio), Program (Program 3), and Instructor (TANIA TURRUBIATES LOPEZ). To the right, there are fields for Start Date (14-oct.-23), End Date, and Language (Java). The main part of the window is a table with the following columns: ID, PID, Phase, Start, Int., Stop Date and Time, Delta, and Comments. The table contains several rows of data, including phases like PLAN, DLD, CODE, COMPILE, UT, and PM, with corresponding start and stop times and delta values.

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
34	423	PLAN	14/10/2023 06:49:58 p. r	0	14/10/2023 07:12:10 p. r	22.2	Estimacion y planeacion
35	423	DLD	14/10/2023 07:12:11 p. r	0	14/10/2023 08:01:35 p. r	49.4	Diseño del diagrama UML
36	423	CODE	14/10/2023 08:01:36 p. r	734	15/10/2023 09:17:09 a. r	61.6	Coficacion / interrupcion para dormir
37	423	COMPILE	15/10/2023 09:17:10 a. r	0	15/10/2023 09:19:10 a. r	2.0	Compilacion
38	423	UT	15/10/2023 09:19:10 a. r	0	15/10/2023 09:29:21 a. r	10.2	Pruebas
39	423	PM	15/10/2023 09:29:22 a. r	0		0.0	
40	423			0		0.0	

En esta fase final, se lleva a cabo un análisis exhaustivo de nuestro proceso de desarrollo, centrándonos en cada etapa, para evaluar nuestro desempeño y los resultados obtenidos:

ETAPA DE PLANEACION:

Comenzamos con un análisis exhaustivo de la descripción del problema, lo que nos permitió comprender a fondo lo que se nos requería y los requisitos clave para nuestro programa. Basándonos en esta comprensión, desarrollamos una estimación del tiempo necesario para completar el proyecto. Además, se hizo necesario establecer una estructura organizativa sólida para llevar a cabo un análisis de regresión lineal de manera más efectiva. Durante esta etapa, notamos que los programadores tienden a experimentar ansiedad y distracciones con facilidad. Por lo tanto, se recomienda trabajar en un entorno tranquilo y sin interrupciones, lo que contribuirá a mejorar la eficiencia y la concentración en el trabajo.

ETAPA DE DISEÑO:

En esta etapa en base a lo comprendido anteriormente se procedió a elaborar el diseño en base al código realizado en la etapa anterior, para lo cual se tuvo que modificar el código de prueba para poderlo implementar en nuestro nuevo diseño.

ETAPA DE CODIFICACION:

En la fase de codificación, se procedió a implementar el arreglo de datos que habíamos diseñado previamente. A pesar de no contar con un diseño detallado por parte del programador, la codificación transcurrió sin contratiempos. El proceso se desarrolló de manera fluida, y no se presentaron problemas significativos ni necesidad de correcciones adicionales. Esto refleja una planificación efectiva y una ejecución exitosa de la etapa de codificación.

ETAPA DE COMPILACION:

En la fase de compilación, el código se procesó sin inconvenientes. El compilador no generó errores ni advertencias significativas. El proceso de verificación de sintaxis se completó con éxito, lo que confirmó la corrección y coherencia del código fuente. Esto allanó el camino para avanzar hacia la siguiente etapa del desarrollo del software con confianza en la integridad del programa.



REPORTE DE PRÁCTICAS

ETAPA DE PRUEBAS:

En la etapa de pruebas, llevamos a cabo una serie de evaluaciones exhaustivas en nuestro programa. Estas pruebas se realizaron meticulosamente para garantizar su robustez y confiabilidad. A medida que ejecutamos las pruebas, observamos que el programa se comportaba de acuerdo con nuestras expectativas y cumplía con los requisitos establecidos previamente. No se identificaron fallos ni problemas significativos durante las pruebas, lo que indica que el software estaba listo para su implementación y uso. Las pruebas exitosas validaron la calidad y el rendimiento del programa, brindando la tranquilidad de que estaba preparado para su despliegue.

EVALUACION DE RESULTADOS:

METAS ESTABLECIDAS:

En la etapa de planeación se establecieron las metas principales en base a los requerimientos, las principales son:

1. El tiempo estimado para la elaboración de este proyecto fue de 2hr 50 min.
2. El programa debía recibir los datos por teclado.
3. El programa debía implementar listas enlazadas para almacenar y manejar los datos.
4. El programa obtenga y muestre los resultados correctos.

METAS NO CUMPLIDAS:

Una de las metas que nos habíamos fijado no se cumplió completamente en lo que respecta al tiempo. Inicialmente, teníamos planeado que la elaboración del proyecto nos llevaría aproximadamente 2 horas y 50 minutos. La verdad es que traté de trabajar de manera eficiente y dar lo mejor de mí, pero en el proceso surgieron algunas dificultades que no habíamos anticipado. Estas situaciones inesperadas hicieron que el proyecto se extendiera más allá del tiempo estimado. A pesar de ese retraso en el tiempo de desarrollo, me alegra decir que logramos mantener la calidad y la funcionalidad del proyecto tal como se había planeado. Todo está funcionando como debería. Esta experiencia nos servirá de lección para futuros proyectos, ya que nos ayudará a ajustar mejor nuestras estimaciones de tiempo y mejorar la gestión del tiempo en proyectos similares.

METAS CUMPLIDAS:

- El código resultante funciona de acuerdo con los requisitos y expectativas establecidos.
- No se identificaron errores críticos ni problemas de rendimiento.
- Los cálculos de regresión lineal se realizan con precisión, y se logra el propósito previsto del programa.

En general, el proyecto se completó con éxito y todas las metas y objetivos se alcanzaron de manera satisfactoria.



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMPACHE

REPORTE DE PRÁCTICAS

PROPUESTA DE MEJORA DE PROCESOS (PIP)

PSP Process Improvement Proposal

Carnegie Mellon Software Engineering Institute

Student: Gonzalo Martinez Silverio
Program: Program 3
Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 14-oct.-23
End Date: 15-oct.-23
Language: Java

UID: #2Nor Date: 31-oct.-23

Problem Description

Briefly describe the problems you encountered.

Sintaxis:
En numerosos lenguajes de programación, como Java, es necesario utilizar puntos y comas para separar las declaraciones. Olvidar un punto y coma al final de una línea o agregarlo donde no debería estar puede provocar errores, como me ocurrió a mí.

Documentación:
Es fundamental escribir comentarios legibles y coherentes para lograr una comprensión adecuada del código. Esto fue lo que me sucedió, ya que no expliqué de manera precisa a través de comentarios en el

Proposal Description

Briefly describe the process improvements that you propose.

Sintaxis:
Seguir estándares de estilo, utilizando IDEs con resaltado de sintaxis.

Documentación:
Agrega comentarios significativos, documentar funciones, clases.
Aprender continuamente y reflexiona sobre errores pasados.

Other Notes and Comments

Note any other comments or observations that describe your experiences or improvement ideas.

Al seguir recomendaciones y ser diligente en un enfoque hacia la sintaxis y la documentación en la programación, se logrará reducir la probabilidad de errores y mejorar la calidad y la mantenibilidad del código.

PLANTILLA DE ESTIMACIÓN DE TAMAÑO

PSP Size Estimating Template

Carnegie Mellon Software Engineering Institute

Student: Gonzalo Martinez Silverio
Program: Program 3
Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 14-oct.-23
End Date: 15-oct.-23
Language: Java

ID: 423

Parts: Base

ID	Name	Plan				Actual			
		Base	Del.	Mod.	Add	Base	Del.	Mod.	Add
1	Program 1	185	0	0	0	185	62	123	0
vo)		0	0	0	0	0	0	0	0
Base TOTAL		185	0	0	0	185	62	123	0

Parts: Added

ID	Name	Part Type	Plan			Actual		
			Items	Rel. Sz.	Size *	Items	Size *	
21	Formulas	Calc	14		74.0	14	66	
22	Ingresar datos (interfaz consola)	IO	26		28.0	30	29	
vo)			0		0.0	0	0	
Added Parts TOTAL			102			95		



REPORTE DE PRÁCTICAS

Parts: Reused			
ID	Name	Plan	Actual
2	Program 1	4	9
*	svd)	0	0
Reuse Object TOTAL		4	9

PROBE Calculation Worksheet		Size	Time
Added Size (A):	$A=BA+PA$	102	
Estimated A&M (E):	$E=BA+PA+M$	102	
PROBE method used: (A,B,C,D)		D	D
Correlation (R^2):			
Regression Parameter (B0):	Size and Time	0	0
Regression Parameter (B1):	Size and Time	1	1
Projected A&M (P):	$P=B0+B1*E$	102	
Estimated Total Size (T):	$T=P+B-D-M+R$	291	
Estimated Total New Reusable (NR):	sum of * items	0	
Estimated Total Development Time:	$Time=B0+B1*E$		170
Prediction Range:	Range	0	0
Upper Prediction Interval:	$UPI=P+Range$	0	0
Lower Prediction Interval:	$LPI=P-Range$	0	0
Prediction Interval Percent:		70%	70%

PROBE Historical Data - Methods B and C				Plan	Actual			
ID	Name	End Date	Time	A&M	E	Time	A&M	Outlier

PROBE Method Parameters - Methods B and C						
	R^2 Size	Beta0 Size	Beta1 Size	R^2 Time	Beta0 Time	Beta1 Time
Method C						
Method B						

PROBE Historical Data - Method A				Plan	Actual			
ID	Name	End Date	Time	A&M	E	Time	A&M	Outlier

PROBE Method Parameters - Method A						
	R^2 Size	Beta0 Size	Beta1 Size	R^2 Time	Beta0 Time	Beta1 Time
Method A						

Registro: 1 de 1 Sin filtro Buscar



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

REPORTE DE PRÁCTICAS

RESUMEN DEL PLAN DEL PROYECTO

PSP1 Project Plan Summary

Personal Software Process[®]

Carnegie Mellon Software Engineering Institute

Student: Gonzalo Martinez Silverio
Program: Program 3
Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 14-oct.-23
End Date: 15-oct.-23
Language: Java

ID: 4

Summary

	Plan	Actual	To-Date
Productivity	36.0	-2.7	-0.9

Plan Size

	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	-132.00	
Reused (R)	4.00	9.00	9.00
Added & Modified (A&M)	102.00	-9.00	-9.00
Total (T)	291.00	0.00	0.00
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	1	25.0%
CODE		2	3	75.0%
COMPILE		0	0	0.0%
UT		0	0	0.0%
PM		0	0	0.0%
Total		2	4	

Defects Removed in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN		0	0	0.0%
DLD		0	0	0.0%
CODE		2	0	0.0%
COMPILE		0	0	0.0%
UT		0	4	100.0%
PM		0	0	0.0%
Total		2	4	

Registro: 14 1 de 1 Filtrado Buscar



INFORME DE PRUEBA

PSP Test Report	
Test Number	01
Test Name	PROGRAMA 3 PSP1
Objective	Verificar que el código en el archivo RegresionLineal funcione correctamente para calcular la regresión lineal y predecir valores de salida utilizando los datos proporcionados en el archivo ASGKIT PROG3 en el contexto de PSP1
Description	El programa se utiliza para calcular coeficientes de regresión y predecir valores de salida con base en datos proporcionados por el usuario. Este código forma parte del proyecto PROGRAMA 3 y está diseñado para ser una herramienta útil en el proceso PSP1.
Conditions	Se utilizará una entrada de datos de prueba proporcionada por el usuario para cada uno de los cuatro conjuntos de datos. Se asegurará que los datos de entrada cumplan con los requisitos numéricos y de formato esperados.
Expected Results	Para cada conjunto de datos, se esperan los siguientes resultados: Los valores de Beta0 y Beta1 calculados. El coeficiente de correlación (rxy). El coeficiente de determinación (r^2). El valor Yk predicho.
Actual Results	Obtuvimos un exitoso resultado en cada una de las pruebas realizadas en el contexto de PSP1. Los resultados obtenidos coinciden con las expectativas y no se han identificado problemas significativos en la implementación.

Registro: 1 de 1 Sin filtro Buscar



CONCLUSION:

Las etapas de planeación, diseño, codificación, compilación y pruebas se llevaron a cabo de manera eficiente y exitosa en el proyecto. A pesar de enfrentar un retraso en el tiempo estimado para la elaboración, se lograron las metas principales del proyecto. La calidad y la funcionalidad del software se mantuvieron, lo que demuestra una planificación efectiva y una ejecución exitosa. Además, se identificaron áreas de mejora para futuros proyectos, como la gestión del tiempo y las estimaciones de desarrollo.

En este contexto de PSP1, el código ha demostrado ser una herramienta efectiva para realizar cálculos de regresión lineal. Además, el proceso PSP1 ha permitido al programador llevar un seguimiento disciplinado de su trabajo, mejorar su productividad y aplicar buenas prácticas en el desarrollo de software. El éxito de esta prueba valida la utilidad del proceso PSP1 como un enfoque efectivo para la mejora personal del programador y la entrega de código de alta calidad. El código se encuentra en conformidad con estos estándares y está listo para ser utilizado en proyectos futuros. Con lo anterior puedo decir que es importante continuar aplicando los principios del PSP para mantener y mejorar la calidad de trabajo a lo largo de la carrera profesional.

OBSERVACIONES PROPIAS PARA MEJORAR:

1. Mejorar la concentración.
2. Mejorar el tiempo de resolución de un problema.
3. Mejorar los tiempos en cada etapa para ser más eficiente.
4. Evitar interrupciones.

BIBLIOGRAFIA:

Watts S. Humphrey (2005). PSP A Self-Improvement Process for Software Engineers. Addison-Wesley Professional.

PSP Academic Material (2016). Acceso el 20 de Agosto de 2016 desde Team Software Process. Software Engineering Institute, Carnegie Mellon University. Sitio Web: <http://www.sei.cmu.edu/tsp/tools/academic/index.cfm>