



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

CARRERA:

INGENIERIA EN SISTEMAS COMPUTACIONALES

ASIGNATURA:

PROCESO PERSONAL DE DESARROLLO DE SOFTWARE

NOMBRE DE LA PRÁCTICA:

USO DEL PROCESO PSP2.1

NÚMERO DE LA PRÁCTICA:

PROGRAMA 06

ESTUDIANTE:

GONZALO MARTINEZ SILVERIO

DOCENTE:

DRA. TANIA TURRUBIATES LOPEZ

SEMESTRE:

7

GRUPO:

701^a



INTRODUCCIÓN:

El Proceso de Software Personal (PSP) es una metodología que busca mejorar la calidad del software a través de la aplicación de principios disciplinados en el desarrollo de programas. En este reporte, nos centraremos en el uso de PSP2.1, que introduce medidas adicionales para gestionar la calidad del proceso y proporciona plantillas de diseño estructuradas para registrar y documentar diseños de software.

OBJETIVO:

Los objetivos de PSP2.1 son la introducción de medidas adicionales para la gestión de la calidad del proceso y el suministro de plantillas de diseño que ofrecen un marco organizado y un formato estructurado para el registro de diseños. Estos objetivos buscan mejorar la eficiencia y la calidad en el desarrollo de software a nivel personal.

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):

- Computadora personal.
- NetBeans 8.2.
- JDK.
- Material de prácticas de PSP (Requerimientos, Time Log, etc.).

DESARROLLO:

1.- Registramos el inicio de la etapa de planeación en la herramienta PSP Student Workbook.mde:

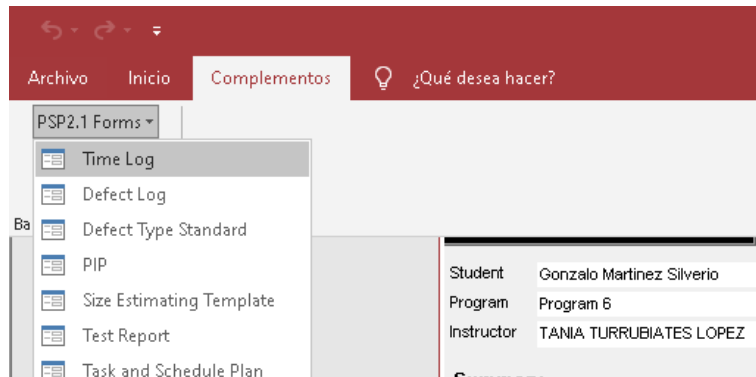
The screenshot shows the 'PSP Student Workbook' application window. It has a menu bar with 'Project Directory', 'Process Directory', 'Size Measures', 'Analysis Tools', 'Scripts/Forms', and 'Course Materials'. Below the menu bar is a toolbar with buttons for 'New Project', 'Open Project', 'Replan Project', 'Delete Project', 'Import', and 'Export'. The main area contains a table with the following data:

ID	Name	Started	Completed
408	Program 1	07/09/2023 07:26:42 p. r	12/09/2023 11:56:59 p. r ✓
422	Program 2	24/09/2023 06:44:14 a. r	26/09/2023 07:17:40 p. r ✓
423	Program 3	14/10/2023 06:46:44 p. r	15/10/2023 10:25:06 a. r ✓
424	Program 4	03/11/2023 06:59:13 a. r	03/11/2023 07:10:58 p. r ✓
425	Program 5	17/11/2023 02:29:52 p. r	24/11/2023 08:13:47 p. r ✓
426	Program 6	15/12/2023 09:00:06 a. r	✓



ETAPA DE PLANEACION:

Comenzamos a registrar en el time log el inicio de la etapa de planeación:



PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Student: Gonzalo Martinez Silverio
Program: Program 6
Instructor: TANIA TURRUBIATES LOPEZ

ID	PID	Phase	Start	Int.	Stop Date and Time
54	426	PLAN	15/12/2023 09:00:18 a.m.	0	
54	426			0	

Requerimientos del programa:

Usando PSP2.1, escriba un programa para encontrar el valor de x para el cual integrar la función t de 0 a x da un resultado de p .

Pruebe minuciosamente el programa. Como mínimo, calcule los valores para la integral de distribución t para los valores de la Tabla 1. Los valores esperados también se incluyen en la Tabla 1.

Prueba		Valor esperado	Valor actual
p	Dof	x	
0,20	6	0.55338	
0,45	15	1,75305	
0.495	4	4.60409	

Tabla 1



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Encuentre el valor de x para el cual integrar la función t de 0 a x da un resultado de p .

- Comience con un valor de prueba para el límite superior de 1 y calcule el valor de la integración.
- Compárelo con el valor deseado.
- Si el resultado de la integración es demasiado bajo, elija un límite superior de prueba mayor
- Si el resultado de la integración es demasiado alto, elija un límite superior de prueba más pequeño

Realice integraciones de prueba sucesivas hasta que el valor de la integración esté dentro de un error aceptable, digamos 0,00001.

Una forma de hacer este cálculo es la siguiente.

Paso	Acción
1	Comience con un valor de prueba de x (por ejemplo, 1,0).
2	Haga una integral inicial y pruebe para ver si da el valor adecuado; si no, continúa.
3	Si es demasiado bajo, agregue $d = 0,5$ a la prueba x .
4	Si es demasiado alto, reste $d = 0,5$ del ensayo x .
5	Integrar nuevamente y probar si el resultado está dentro de un error aceptable; si no, continúa.
6	Si es demasiado bajo, ajuste d ; agregue d a la prueba x .
7	Si es demasiado alto, ajuste d ; restar d del ensayo x .
8	Reciclar a las 5.

Las reglas para ajustar d son estas.

- Mientras las pruebas para detectar el error del resultado den el mismo signo de error, deje d sin cambios.
- Siempre que cambie el signo del error, divida d entre 2.

Tenga en cuenta que este método de ajustar d podría dar como resultado un valor de prueba de $x = 0$.

Para protegerse contra un problema con el método de Simpson, asegúrese de que el programa maneje un valor 0 de la función que se está integrando.

2.-Estimación de tiempo:

El tiempo estimado para este proyecto será de 2 horas y 30 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 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 6), and 'Instructor' (TANIA TURRUBIATES LOPEZ). To the right, there are fields for 'Start Date' (15-dic.-23), 'End Date', and 'Language' (Java). The main table has columns: ID, PID, Phase, Start, Int., Stop Date and Time, Delta, and Comments. The table contains two rows: one for 'PLAN' (ID 54, PID 426) and one for 'DLD' (ID 55, PID 426). The 'PLAN' row shows a start time of 15/12/2023 09:00:18 a. r and a stop time of 15/12/2023 09:10:43 a. r, with a delta of 10.4. The 'DLD' row shows a start time of 15/12/2023 09:10:46 a. r and a delta of 0.0.

ETAPA DE DISEÑO:

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

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 6), and 'Instructor' (TANIA TURRUBIATES LOPEZ). The main table has columns: ID, PID, Phase, Start, Int., Stop Date and Time, and Comments. The table contains two rows: one for 'PLAN' (ID 54, PID 426) and one for 'DLD' (ID 55, PID 426). The 'PLAN' row shows a start time of 15/12/2023 09:00:18 a. r and a stop time of 15/12/2023 09:10:43 a. r, with a delta of 0. The 'DLD' row shows a start time of 15/12/2023 09:10:46 a. r and a delta of 0.

2.- En este diagrama reutilizamos métodos del programa anterior a esta práctica, el diseño se presentará será la siguiente manera: Los métodos y atributos privados están representados con el signo -. Los métodos públicos están representados con el signo +. El método main es público y está destinado a ser el punto de entrada del programa. Los métodos privados como `distribucionT`, `integrarDistribucionT`, `calcularValorIntegral`, y `encontrarXParaProbabilidad` son utilizados internamente para realizar cálculos específicos y no son accesibles desde fuera de la clase PROGRAMA6.

```
+-----+
|          PROGRAMA6          |
+-----+
| - distribucionT(x, dof)      |
| - integrarDistribucionT(a, b, dof, E) |
| - calcularValorIntegral(h, a, dof, num_seg) |
| - encontrarXParaProbabilidad(p, dof, E) |
| + main(String[] args)      |
+-----+
```



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

3.-Finalizando la etapa de Diseño lo registramos en el Time Log:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Personal Software Process[®]

Student: Gonzalo Martinez Silverio

Program: Program 6

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23

End Date:

Language: Java

ID: 426

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0		0.0	

ETAPA DE REVISION DE DISEÑO

3.-Iniciamos la etapa de Revisión de Diseño y lo registramos en el Time Log:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Personal Software Process[®]

Student: Gonzalo Martinez Silverio

Program: Program 6

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23

End Date:

Language: Java

ID: 426

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0		0.0	Revision del diseño del diagrama

Finalizamos la etapa de revisión de diseño y lo registramos en el Time Log.

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Personal Software Process[®]

Student: Gonzalo Martinez Silverio

Program: Program 6

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23

End Date:

Language: Java

ID: 426

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revision del diseño del diagrama



ETAPA DE CODIFICACION:

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

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revisión del diseño del diagrama
57	426	CODE	15/12/2023 09:54:00 a. r	0		0.0	Codificación
58	426			0		0.0	

Abrimos NetBeans 8.2 nos dirigimos a File > New Project > Seleccionamos Java en Categories > Java Application en Projects > hacemos clic en Next > ingresamos el nombre el proyecto > creamos la clase llamada PROGRAMA6 > hacemos clic derecho > Create Main Class > hacemos clic en Finish y comenzamos a codificar nuestro programa conforme a nuestro diseño.

```
1 package PROGRAMA6;
2 /**
3  * @author Gonzalo Martinez Silverio
4  */
5 import java.util.Scanner;
6 import org.apache.commons.math3.distribution.TDistribution;
7
8 public class PROGRAMA6 {
9
10    // Reutilizando los métodos existentes de la distribución t
11    public static double distribucionT(double x, int dof) {
12        // Crear un objeto de la clase TDistribution con el grado de libertad dado
13        TDistribution tDistribution = new TDistribution(dof);
14        // Devolver la densidad de la distribución t en el punto x
15        return tDistribution.density(x);
16    }
17
18    // Reutilizando los métodos existentes de la integrarDistribucionT
19    public static double integrarDistribucionT(double a, double b, int dof, double E) {
20        // Inicializar el número de segmentos con 10
21        int num_seg = 10;
22        // Calcular el ancho de cada segmento
23        double h = (b - a) / num_seg;
24        // Calcular el valor inicial de la integral usando la regla de Simpson
25        double valorIntegral = calcularValorIntegral(h, a, dof, num_seg);
26    }
27 }
```



```
25
26 // Repetir el proceso hasta que el error sea menor que el epsilon dado
27 while (true) {
28     // Duplicar el número de segmentos
29     num_seg *= 2;
30     // Recalcular el ancho de cada segmento
31     h = (b - a) / num_seg;
32     // Calcular el nuevo valor de la integral usando la regla de Simpson
33     double nuevoValorIntegral = calcularValorIntegral(h, a, dof, num_seg);
34
35     // Verificar si la diferencia entre el nuevo y el viejo valor es menor que el epsilon
36     if (Math.abs(nuevoValorIntegral - valorIntegral) < E) {
37         // Salir del bucle
38         break;
39     }
40
41     // Actualizar el valor de la integral con el nuevo valor
42     valorIntegral = nuevoValorIntegral;
43 }
44
45 // Devolver el valor final de la integral
46 return valorIntegral;
47 }
48 // Reutilizando los métodos existentes de la calcularValorIntegral
49 private static double calcularValorIntegral(double h, double a, int dof, int num_seg) {
50     // Inicializar el valor de la integral con 0
51     double valorIntegral = 0.0;
52     // Declarar una variable para almacenar el valor de x en cada segmento
53     double x;
54
55     // Recorrer todos los segmentos
56     for (int i = 0; i <= num_seg; i++) {
57         // Calcular el valor de x en el segmento i
58         x = a + i * h;
59         // Aplicar la fórmula de Simpson según el caso
60         if (i == 0 || i == num_seg) {
61             // Si es el primer o el último segmento, multiplicar por 1
62             valorIntegral += distribucionT(x, dof);
63         } else if (i % 2 == 0) {
64             // Si es un segmento par, multiplicar por 2
65             valorIntegral += 2 * distribucionT(x, dof);
66         } else {
67             // Si es un segmento impar, multiplicar por 4
68             valorIntegral += 4 * distribucionT(x, dof);
69         }
70     }
71
72     // Multiplicar el valor de la integral por el ancho de cada segmento dividido por 3
73     valorIntegral *= h / 3;
74
75     // Devolver el valor de la integral
76     return valorIntegral;
77 }
```




INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

```
78 // Nuevo método para encontrar x para una probabilidad dada por el usuario
79 public static double encontrarXParaProbabilidad(double p, int dof, double E) {
80     // Inicializar x con una suposición de 1
81     double xAdivinanza = 1.0;
82     // Inicializar d con un valor de ajuste de 0.51
83     double d = 0.51;
84
85     // Repetir el proceso hasta encontrar el valor de x que cumpla con la probabilidad deseada
86     while (true) {
87         // Calcular la probabilidad actual integrando la distribución t desde 0 hasta x
88         double pActual = integrarDistribucionT(0, xAdivinanza, dof, E);
89
90         // Verificar si el resultado actual está dentro del rango de error aceptable
91         if (Math.abs(pActual - p) < E) {
92             // Salir del bucle
93             break;
94         }
95
96         // Ajustar x en función de la comparación con la probabilidad objetivo
97         if (pActual < p) {
98             // Si la probabilidad actual es menor que la deseada, aumentar x
99             xAdivinanza += d;
100         } else {
101             // Si la probabilidad actual es mayor que la deseada, disminuir x
102             xAdivinanza -= d;
103         }
104     }
105
106     // Realizar otra integración y verificar el resultado
107     double nuevaP = integrarDistribucionT(0, xAdivinanza, dof, E);
108
109     // Ajustar d en función del cambio en el signo del error
110     if (Math.signum(nuevaP - p) != Math.signum(pActual - p)) {
111         // Si el signo del error cambia, reducir d a la mitad
112         d /= 2.0;
113     }
114
115     // Devolver el valor de x encontrado
116     return xAdivinanza;
117 }
118
119 // Reutilizando y modificado el método main existente
120 public static void main(String[] args) {
121     // Crear un objeto de la clase Scanner para leer la entrada del usuario
122     Scanner scanner = new Scanner(System.in);
123     // Crear una matriz de 3x3 para almacenar los resultados
124     double[][] resultados = new double[3][3];
125
126     // Pedir al usuario que ingrese la probabilidad y el grado de libertad para cada caso
127     for (int i = 0; i < 3; i++) {
128         System.out.print("Ingrese la probabilidad deseada (p): ");
129         double probabilidadObjetivo = scanner.nextDouble();
130         System.out.print("Ingrese el valor (dof) para x[" + i + "]: ");
131         int dof = scanner.nextInt();
132         // Llamar al método encontrarXParaProbabilidad para obtener el valor de x correspondiente
133         double xParaProbabilidad = encontrarXParaProbabilidad(probabilidadObjetivo, dof, 0.00001);
134
135         // Almacenar el resultado en la matriz
136         resultados[i][0] = probabilidadObjetivo;
137         resultados[i][1] = dof;
138         resultados[i][2] = xParaProbabilidad;
139     }
140
141     // Imprimir los resultados en forma de tabla
142     System.out.println("\nRESULTADOS");
143     System.out.printf("%10s | %10s | %10s\n", "p", "dof", "Resultado");
144     for (int i = 0; i < 3; i++) {
145         System.out.printf("%10.3f | %10d | %10.5f\n", resultados[i][0], (int) resultados[i][1], resultados[i][2]);
146     }
147     // Cerrar el objeto Scanner
148     scanner.close();
149 }
```



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

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

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Personal Software ProcessSM

PSP Time Recording Log

ID 426

Student: Gonzalo Martinez Silverio

Program: Program 6

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23

End Date:

Language: Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revisión del diseño del diagrama
57	426	CODE	15/12/2023 09:54:00 a. r	0	15/12/2023 10:24:10 a. r	30.2	Codificación
58	426	CR	15/12/2023 10:24:12 a. r	0	15/12/2023 10:34:47 a. r	10.6	Revisión del código

ETAPA DE REVISIÓN DE CODIGO:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Personal Software ProcessSM

PSP Time Recording Log

ID 426

Student: Gonzalo Martinez Silverio

Program: Program 6

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23

End Date:

Language: Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revisión del diseño del diagrama
57	426	CODE	15/12/2023 09:54:00 a. r	0	15/12/2023 10:24:10 a. r	30.2	Codificación
58	426	CR	15/12/2023 10:24:12 a. r	0	15/12/2023 10:34:47 a. r	10.6	Revisión del código

Marcamos finalizado la etapa de revisión de código en el Time Log.

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Personal Software ProcessSM

PSP Time Recording Log

ID 426

Student: Gonzalo Martinez Silverio

Program: Program 6

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23

End Date:

Language: Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revisión del diseño del diagrama
57	426	CODE	15/12/2023 09:54:00 a. r	0	15/12/2023 10:24:10 a. r	30.2	Codificación
58	426	CR	15/12/2023 10:24:12 a. r	0	15/12/2023 10:34:47 a. r	10.6	Revisión del código



ETAPA DE COMPILACION:

Marcamos el inicio y final de la etapa de compilación:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Personal Software Process

Student: Gonzalo Martinez Silverio

Program: Program 6

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23

End Date:

Language: Java

ID: 426

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revision del diseño del diagrama
57	426	CODE	15/12/2023 09:54:00 a. r	0	15/12/2023 10:24:10 a. r	30.2	Codificación
58	426	CR	15/12/2023 10:24:12 a. r	0	15/12/2023 10:34:47 a. r	10.6	Revision del código
60	426	COMPILE	15/12/2023 10:34:49 a. r	0	15/12/2023 10:35:50 a. r	1.0	Compilamos el programa
60	426	UT	15/12/2023 10:35:51 a. r	0	15/12/2023 10:35:51 a. r	0.0	Pruebas con tabla 1

ETAPA DE TESTEO:

Registramos el inicio de nuestra etapa de Testeo:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute

Personal Software Process

Student: Gonzalo Martinez Silverio

Program: Program 6

Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23

End Date:

Language: Java

ID: 426

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revision del diseño del diagrama
57	426	CODE	15/12/2023 09:54:00 a. r	0	15/12/2023 10:24:10 a. r	30.2	Codificación
58	426	CR	15/12/2023 10:24:12 a. r	0	15/12/2023 10:34:47 a. r	10.6	Revision del código
60	426	COMPILE	15/12/2023 10:34:49 a. r	0	15/12/2023 10:35:50 a. r	1.0	Compilamos el programa
61	426	UT	15/12/2023 10:35:51 a. r	0	15/12/2023 10:35:51 a. r	0.0	Pruebas con tabla 1



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Introducimos los datos de la tabla 1:

```
Output - PSP (run) X
run:
Ingrese la probabilidad deseada (p): 0.20
Ingrese el valor (dof) para x[0]: 6
Ingrese la probabilidad deseada (p): 0.45
Ingrese el valor (dof) para x[1]: 15
Ingrese la probabilidad deseada (p): 0.495
Ingrese el valor (dof) para x[2]: 4

RESULTADOS
| p | dof | Resultado
| 0.200 | 6 | 0.55338
| 0.450 | 15 | 1.75305
| 0.495 | 4 | 4.60188
BUILD SUCCESSFUL (total time: 24 seconds)
```

Los resultados fueron verificados y coinciden por lo tanto marcamos como finalizada la etapa de pruebas:

</



ETAPA POSMORTEM:

Registramos el inicio de esta etapa:

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revisión del diseño del diagrama
57	426	CODE	15/12/2023 09:54:00 a. r	0	15/12/2023 10:24:10 a. r	30.2	Codificación
58	426	CR	15/12/2023 10:24:12 a. r	0	15/12/2023 10:34:47 a. r	10.6	Revisión del código
60	426	COMPILE	15/12/2023 10:34:49 a. r	0	15/12/2023 10:35:50 a. r	1.0	Compilamos el programa
61	426	UT	15/12/2023 10:35:51 a. r	0	15/12/2023 10:38:49 a. r	3.0	Pruebas con tabla 1
62	426	PM	15/12/2023 10:38:50 a. r	0		0.0	

En esta fase llevamos a cabo un exhaustivo análisis de nuestro proceso de desarrollo, centrándonos en cada etapa para evaluar nuestro desempeño y los resultados obtenidos de manera detallada.

ETAPA DE POST MORTEM

ETAPA DE PLANEACION:

La planificación de este proyecto se inició con un análisis detallado de la descripción del problema planteado en el ASGKIT PROG6. Una vez comprendiendo los requerimientos del programa, se realizó una estimación del tiempo necesario para completar el proyecto de manera efectiva.

ETAPA DE DISEÑO:

En esta etapa en base a lo comprendido en la etapa de planeación se procedió a elaborar el diseño en base a los requerimientos, para lo cual se tuvo que diseñar un diagrama de clases en el cual nos dimos cuenta que se podría reutilizar ésta el diseño de la practica anterior a esta..

ETAPA DE CODIFICACION:

Durante la fase de codificación, implementamos una estrategia eficiente al reutilizar código proveniente de la práctica anterior. Este enfoque no solo redujo significativamente el tiempo necesario para completar la codificación, sino que



también ofreció ventajas adicionales para el desarrollo del proyecto. La reutilización de código no solo agilizó el proceso, sino que también brindó beneficios en términos de consistencia y confiabilidad. Al incorporar componentes ya probados y optimizados, pudimos construir sobre una base sólida, minimizando la posibilidad de errores y facilitando la integración de nuevas funcionalidades.

ETAPA DE COMPILACION:

En esta etapa de compilación duro un tiempo mínimo casi inexistente debido a que no se encontraron errores de sintaxis.

ETAPA DE PRUEBAS:

En esta etapa de pruebas identifiqué un error introducido durante la fase de codificación y lo corregí durante las pruebas. El error estaba relacionado con una función que no recibía un parámetro necesario.

EVALUACION DE RESULTADOS:

METAS ESTABLECIDAS:

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

1. El tiempo estimado para la elaboración de este proyecto fue de 2horas y 30 minutos.
2. El programa debía de integrar numéricamente una función usando la regla de Simpson. Utilizar la distribución t como función
3. Calcular los valores para la integral de distribución t para los valores de la tabla 1.
4. El programa obtenga y muestre los resultados correctos.

.

METAS NO CUMPLIDAS:

Todas las metas fueron cumplidas por lo cual no hubo ningún inconveniente.

METAS CUMPLIDAS:

Gracias a la gran comprensión de los requerimientos y a un buen desempeño durante cada una de las etapas y un gran conocimiento se pudo cumplir todas las metas establecidas. Los datos mostrados son confiables ya que el programa respeta cada operación de las fórmulas correspondientes.



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Una vez terminada esta etapa procedemos a registrarla en el time Log y procedemos a terminar el proceso.

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
54	426	PLAN	15/12/2023 09:00:18 a. r	0	15/12/2023 09:10:43 a. r	10.4	Planeacion y analisis de requerimientos
55	426	DLD	15/12/2023 09:10:46 a. r	0	15/12/2023 09:41:55 a. r	31.2	Diseño del diagrama UML
56	426	DLDR	15/12/2023 09:42:00 a. r	0	15/12/2023 09:53:58 a. r	12.0	Revisión del diseño del diagrama
57	426	CODE	15/12/2023 09:54:00 a. r	0	15/12/2023 10:24:10 a. r	30.2	Codificación
58	426	CR	15/12/2023 10:24:12 a. r	0	15/12/2023 10:34:47 a. r	10.6	Revisión del código
60	426	COMPILE	15/12/2023 10:34:49 a. r	0	15/12/2023 10:35:50 a. r	1.0	Compilamos el programa
61	426	UT	15/12/2023 10:35:51 a. r	0	15/12/2023 10:38:49 a. r	3.0	Pruebas con tabla 1
62	426	PM	15/12/2023 10:38:50 a. r	0	15/12/2023 10:55:23 a. r	16.6	Post mortem
63	426			0		0.0	

REGISTRO DE DEFECTOS


Debido a un buen diseño, reutilización de código y a la experiencia del programador no se obtuvo errores.

ID	PID	Date	Type	Phase Injected	Phase Removed	Fix Time	FixDefect
63	426					0	



PROPUESTA DE MEJORA DEL PROCESO

PSP Process Improvement Proposal

**Carnegie Mellon
Software Engineering Institute**

PSP Process Improvement Proposal

StudentGonzalo Martinez Silverio

ProgramProgram 6

InstructorTANIA TURRUBIATES LOPEZ

Start Date15-dic.-23

End Date

LanguageJava

UID#2NorDate15-dic.-23

Problem Description

Briefly describe the problems you encountered.

Debido a un buen diseño, reutilización de código y a la experiencia del programador no se obtuvo errores.

Proposal Description

Briefly describe the process improvements that you propose.

Como desarrollador propongo tener en cuenta la revision del diseño y la revision de codigo ya que si ayuda dal tener menos errores o incluso ningun error.

Other Notes and Comments

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

Con la experiencia en PSP me pude dar cuenta que es necesario para tener un control al momento de desarrollar software debido a que permite mejorar en algunos aspectos que no tomaba en cuenta como lo es el diseño.



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

PLANTILLA DE ESTIMACIÓN DE TAMAÑO

PSP Size Estimating Template

Carnegie Mellon Software Engineering Institute

Personal Software Process[®]

PSP Size Estimating Template

ID 426

Student: Gonzalo Martinez Silverio Start Date: 15-dic.-23

Program: Program 6 End Date:

Instructor: TANIA TURRUBIATES LOPEZ Language: Java

Parts: Base

ID	Name	Plan				Actual			
		Base	Del.	Mod.	Add	Base	Del.	Mod.	Add
7	Program 5	76	0	0	0	76	0	19	18
vo)		0	0	0	0	0	0	0	0
Base TOTAL		76	0	0	0	76	0	19	18

Parts: Added

ID	Name	Part Type	Plan		Actual	
			Items	Size *	Items	Size *
32	encontrarXParaProbabilidad	Calc	5	VS 11.7	14	15
vo)			0	0.0	0	0
Added Parts TOTAL			11.7		15	

PROBE Calculation Worksheet

		Size	Time
Added Size (A):	$A=BA+PA$	11.7	
Estimated A&M (E):	$E=BA+PA+M$	11.7	
PROBE method used: (A,B,C,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):	$P=B0+B1E$	11.7	
Estimated Total Size (T):	$T=P+B-D-M+R$	101.7	
Estimated Total New Reusable (NR):	sum of * items	0	
Estimated Total Development Time:	$Time=B0+B1E$		150
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

ID	Name	End Date	Time	Plan		Actual		Outlier
				A&M	E	Time	A&M	
+02	Program 4	11/03/23	240	58	58	129.37	51	<input type="checkbox"/>
+02	Program 5	11/24/23	125	32.36	32.36	123.85	53	<input type="checkbox"/>

PROBE Method Parameters - Methods B and C

	R^2 Size	Beta0 Size	Beta1 Size	R^2 Time	Beta0 Time	Beta1 Time
Method C	N/A	0	1.150952	N/A	0	2.8023093
Method B	0	0	0	0	0	0

PROBE Historical Data - Method A

ID	Name	End Date	Time	Plan		Actual		Outlier
				A&M	E	Time	A&M	
+02	Program 4	11/03/23	240	58	58	129.37	51	<input type="checkbox"/>
+02	Program 5	11/24/23	125	32.36	32.36	123.85	53	<input type="checkbox"/>

PROBE Method Parameters - Method A

	R^2 Size	Beta0 Size	Beta1 Size	R^2 Time	Beta0 Time	Beta1 Time
Method A	0	0	0	0	0	0

Registro: 1 de 1

Sin filtro

Buscar



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

INFORME DE PRUEBA

PSP Test Report

Carnegie Mellon Software Engineering Institute

Student: Gonzalo Martínez Silverio
Program: Program 6
Instructor: TANIA TURRUBIATES LOPEZ

Start Date: 15-dic.-23
End Date:
Language: Java

Test Number: 01
Test Name: Prueba tabla 1
Objective: Obtener 0 errores y obtener resultados similares a la tabla 1


Description: Probar minuciosamente el programa. Como mínimo, calcule los valores para la integral de distribución t para los valores de la Tabla 1

Conditions:
- Si el resultado de la integración es demasiado bajo, elija un límite superior de prueba mayor
- Si el resultado de la integración es demasiado alto, elija un límite superior de prueba más pequeño

Expected Results: Los valores esperados son
0.55338
1.75305
4.60409

Actual Results: Los valores actuales son similares a los esperados.
0.55338
1.75305
4.60188

PLANES DE TAREAS Y HORARIOS



Personal Software Processes, Inc.

Software Engineering Institute

PSP Task and Schedule Plans

Task and Schedule Plan Status

Student

Gonzalo Martinez Silverio

Start Date

28-mar.-05

Instructor

TANIA TURRUBIATES LOPEZ

Default Hours/Week

10

Task & Schedule Plan

My Task and Schedule Plan

Planned Weeks

0

ID:

1

Total Plan Hours

0

Task Plan

Schedule Plan

Cumulative Earned Value

Cumulative Hours

Task					Plan		Actual			
ID	Order	Project	Phase	Task Name	Hours	Date	PV	Hours	Date	EV
	745	19	Program 4	PLAN	0.33		0.00	0.3		0.00
	746	20	Program 4	DLD	1.20		0.00	0.5		0.00
	747	21	Program 4	CODE	0.90		0.00	0.6		0.00
	748	22	Program 4	COMPILE	0.01		0.00	0.0		0.00
	749	23	Program 4	UT	0.31		0.00	0.2		0.00
	750	24	Program 4	PM	1.24		0.00	0.5		0.00
	751	25	Program 5	PLAN	0.00		0.00	0.3		0.00
	752	26	Program 5	DLD	0.00		0.00	0.5		0.00
	753	27	Program 5	DLDL	0.00		0.00	0.0		0.00
	754	28	Program 5	CODE	0.00		0.00	0.7		0.00
	755	29	Program 5	CR	0.00		0.00	0.0		0.00
	756	30	Program 5	COMPILE	0.00		0.00	0.0		0.00
	757	31	Program 5	UT	0.00		0.00	0.1		0.00
	758	32	Program 5	PM	0.00		0.00	0.6		0.00
	759	33	Program 6	PLAN	0.30		0.00	0.2		0.00
	760	34	Program 6	DLD	0.58		0.00	0.5		0.00
	761	35	Program 6	DLDL	0.00		0.00	0.2		0.00
	762	36	Program 6	CODE	0.84		0.00	0.5		0.00
	763	37	Program 6	CR	0.00		0.00	0.2		0.00
	764	38	Program 6	COMPILE	0.02		0.00	0.0		0.00
	765	39	Program 6	UT	0.08		0.00	0.0		0.00
	766	40	Program 6	PM	0.67		0.00	0.3		0.00

Registro:

1 de 1

Sin filtro

Buscar

RESUMEN DEL PLAN DE PROTECCIÓN

PSP2.1 Project Plan Summary

Carnegie Mellon
Software Engineering Institute

Personal Software ProcessSM
PSP2.1 Project Plan Summary

Student
Gonzalo Martinez Silverio

Program
Program 6

Instructor
TANIA TURRUBIATES LOPEZ

Start Date
15-dic.-23

End Date

Language
Java

ID
426

Summary

	Plan	Actual	To-Date
Productivity	4.7	30.8	54.3
Planned Time	150.0		1125.0
Actual Time		114.8	1195.0
CPI			0.94
%Reused	13.8	21.1	12.6
%New Reusable	0.0	0.0	0.0
Test Defects/KLOC	18.9	0.0	8.9
Total Defects/KLOC	18.9	0.0	8.9
Yield%	0.0	0.0	0.0
%Appraisal COQ	0.0	19.6	0.0
%Failure COQ	4.3	3.5	4.3
COQ A/F Ratio	0.0	5.7	0.0

Program Size Summary

LOC-Lines of code

	Plan Size	Actual Size	To Date
Base (B)	76.00	76.00	
Deleted (D)	0.00	0.00	
Modified (M)	0.00	19.00	
Added (A)	11.70	40.00	
Reused (R)	14.00	31.00	35.00
Added & Modified (A&M)	11.70	59.00	112.00
Total (T)	101.70	147	277.00
New Reusable (NR)	0.00	0.00	0.00
Estimated A&M (E)	11.70		
UPI	0.00		
LPI	0.00		

Time in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN	18	10	15	12.2%
DLD	35	31	29	23.3%
DLDR	0	12	0	0.0%
CODE	50	30	42	33.5%
CR	0	11	0	0.0%
COMPILE	1	1	1	0.9%
UT	5	3	4	3.4%
PM	40	17	33	26.8%
Total	150	115	124	
UPI	0			
LPI	0			



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

PSP2.1 Project Plan Summary

Carnegie Mellon Software Engineering Institute

Personal Software ProcessSM

PSP2.1 Project Plan Summary

Defects Injected in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN	0.0	0	0	0.0%
DLD	0.0	0	0	0.0%
DLDR	0.0	0	0	0.0%
CODE	0.2	0	1	100.0%
CR	0.0	0	0	0.0%
COMPILE	0.0	0	0	0.0%
UT	0.0	0	0	0.0%
PM	0.0	0	0	0.0%
Total	0.2208	0	1	

Defects Removed in Phase

Phase	Plan	Actual	To-Date	To-Date%
PLAN	0.0	0	0	0.0%
DLD	0.0	0	0	0.0%
DLDR	0.0	0	0	0.0%
CODE	0.0	0	0	0.0%
CR	0.0	0	0	0.0%
COMPILE	0.0	0	0	0.0%
UT	0.2	0	1	100.0%
PM	0.0	0	0	0.0%
Total	0.2208	0	1	

Defect Removal Efficiency

	Plan	Actual	To-Date
Defects/Hr. DLDR	0.0	0.0	0.0
Defects/Hr. CR	0.0	0.0	0.0
Defects/Hr. Compile	0.0	0.0	0.0
Defects/Hr. UT	2.6	0.0	8.4
DRL (DLDR/UT)	0.0	0.0	0.0
DRL (CR/UT)	0.0	0.0	0.0
DRL (Compile/UT)	0.0	0.0	0.0

Registro: 1 de 1 | Filtrado | Buscar

Registramos el final del proyecto.

Main

Carnegie Mellon Software Engineering Institute

Personal Software ProcessSM

PSP Student Workbook

Project Directory | Process Directory | Size Measures | Analysis Tools | Scripts/Forms | Course Materials

Projects

New Project | Open Project | Replan Project | Delete Project | Import Export

ID	Name	Started	Completed
408	Program 1	07/09/2023 07:26:42 p. r	12/09/2023 11:56:59 p. r
422	Program 2	24/09/2023 06:44:14 a. r	26/09/2023 07:17:40 p. r
423	Program 3	14/10/2023 06:46:44 p. r	15/10/2023 10:25:06 a. r
424	Program 4	03/11/2023 06:59:13 a. r	03/11/2023 07:10:58 p. r
425	Program 5	17/11/2023 02:29:52 p. r	24/11/2023 08:13:47 p. r
426	Program 6	15/12/2023 09:00:06 a. r	15/12/2023 10:56:03 a. r



CONCLUSIONES

Al finalizar este proyecto sobre el uso de PSP2.1, se destaca claramente el impacto significativo que esta metodología puede tener en la gestión y mejora de la calidad en el proceso de desarrollo de software personal. PSP2.1 no solo proporciona un marco estructurado, sino que también integra herramientas específicas y enfoques que se traducen en un desarrollo más eficiente y robusto.

La introducción de medidas adicionales para gestionar la calidad del proceso, junto con la implementación de plantillas de diseño, demuestra un compromiso firme con la excelencia y la estandarización en cada fase del ciclo de vida del desarrollo de software. La inclusión de elementos la lista de verificación para la revisión de diseño, así como las diversas plantillas de especificaciones, agrega una capa adicional de rigurosidad y detalle que contribuye a la coherencia y la calidad del producto final.

La automatización de los intervalos de predicción del tamaño del programa, en combinación con el enfoque COQ (Costo de la Calidad), proporciona un marco cuantificable para evaluar y gestionar la inversión en calidad. Esto no solo permite una planificación más precisa, sino que también sienta las bases para una toma de decisiones informada y basada en datos durante el desarrollo.

El énfasis en la relación COQ A/F destaca la importancia estratégica de equilibrar los costos de evaluación con los costos de falla, promoviendo un enfoque proactivo hacia la identificación y corrección de problemas en etapas tempranas del proceso.



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Plantilla de especificación operativa

Alumno: Gonzalo Martínez Silverio

Fecha: 15 / 12 / 2023

Programa: ASGKIT PROG6

Programa: #5

Instructor: Dra. Tania Turrubiates López

Lenguaje: Java

Número de escenario	01	Objetivo del usuario	Ingresar probabilidad y grados de libertad
Objetivo del escenario		El usuario desea obtener el valor de x para una probabilidad específica y grados de libertad.	
Fuente	Paso	Acción	Comentarios
Usuario	1	Ingresar la probabilidad deseada (p).	Asegurarse de que la probabilidad sea no negativa.
Usuario	2	Ingresar el valor de grados de libertad (dof)	Asegurarse de que los grados de libertad sean positivos.
Usuario	3	Llamar al método encontrarXParaProbabilidad.	
Programa	1	Inicializar xAdivinanza con 1.0 y d con 0.51.	
Programa	2	Iterar hasta encontrar el valor de x que cumpla con la probabilidad deseada.	
Programa	2.1	Calcular la probabilidad actual integrando la distribución t desde 0 hasta xAdivinanza.	
Programa	2.2	Verificar si el resultado actual está dentro del rango de error aceptable.	
Programa	2.3	Ajustar x en función de la comparación con la probabilidad objetivo.	
Programa	2.4	Realizar otra integración y verificar el resultado.	
Programa	2.5	Ajustar d en función del cambio en el signo del error.	
Programa	3	Devolver el valor de xAdivinanza	
Usuario	4	Almacenar el resultado en la matriz resultados.	
Usuario	5	Imprimir los resultados en forma de tabla.	
Programa	5.1	Cerrar el objeto Scanner	



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Plantilla de especificación funcional

Alumno: Gonzalo Martínez Silverio

Fecha: 15 / 12 / 2023

Programa: ASGKIT PROG6

Programa: #5

Instructor: Dra. Tania Turrubiates López

Lenguaje: Java

Nombre de la clase	PROGRAMA6
Clase para padres	Solo existe una clase.

Atributos / Elementos	
Declaración	Descripción
Variables y Parámetros: 1. xAdivinanza 2. d 3. p 4. dof 5. E	1. Suposición inicial para encontrar el valor de x. 2. Valor de ajuste utilizado para modificar x. 3. Parámetro: Probabilidad deseada. 4. Parámetro: Grado de libertad. 5. Parámetro: Error aceptable.
Métodos: - distribucionT(double x, int dof) - integrarDistribucionT(...) - calcularValorIntegral(...) - encontrarXParaProbabilidad(...)	Calcula la densidad de la distribución t en el punto x. Realiza la integración numérica de la distribución t. Calcula el valor de la integral utilizando la regla de Simpson. Encuentra el valor de x para una probabilidad dada.
Matriz resultados	Almacena resultados de cálculos (3x3): probabilidad, dof, x.
Entrada/Salida - Scanner - System.out.println(...)	Objeto para leer entrada del usuario. Imprime resultados y mensajes en la consola.
Bucles y Estructuras de Control - while(true) - for(int i = 0; i < 3; - if, else	Bucle para ajustar x hasta que se cumple cierta condición. Bucle para ingresar probabilidad y dof tres veces. Estructuras condicionales para tomar decisiones.
Manejo de Errores - No hay manejo explícito de excepciones.	Se utiliza while(true) para ajustar hasta que se cumpla cierta condición.



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Plantilla de especificación estatal

Alumno: Gonzalo Martínez Silverio

Fecha: 15 / 12 / 2023

Programa: ASGKIT PROG6

Programa: #5

Instructor: Dra. Tania Turrubiates López

Lenguaje: Java

Nombre del Estado	Descripción	
Función/Parámetro	Descripción	
Estados/Próximos	Estados Condición de transición	Acción



INSTITUTO TECNOLÓGICO SUPERIOR DE ALAMO TEMAPACHE

Plantilla de especificación lógica

Alumno: Gonzalo Martínez Silverio

Fecha: 15 / 12 / 2023

Programa: ASGKIT PROG6

Programa: #5

Instructor: Dra. Tania Turrubiates López

Lenguaje: Java

Referencias de diseño: _____

Parámetros: _____