INSTITUTO TECNOLOGICO SUPERIOR DE ALAMO TEMAPACHE

CARRERA:

INGENIERIA EN SISTEMAS COMPUTACIONALES

ASIGNATURA:

PROCESO PERSONAL DE SOFTWARE (PSP)

NOMBRE DE LA PRÁCTICA:

USO DE PSP0.0

INTEGRANTES DEL EQUIPO:

GONZALO MARTINEZ SILVERIO

DOCENTE:
DRA. TANIA TURRUBIATES LOPEZ

SEMESTRE:

7

GRUPO:

701A

INTRODUCCIÓN:

Resumen: El Proceso Software Personal (Personal Software Process o PSP) es un proceso individual cuyo objetivo es ayudar a los ingenieros en software a medir y mejorar su productividad personal. El PSP suele enseñarse exclusivamente a profesionales, no obstante, éste ha comenzado a incluirse como parte de cursos universitarios. En este trabajo se presenta un estudio sobre PSP efectuado en un entorno académico donde se analizan los siguientes indicadores: precisión en las estimaciones de tamaño y esfuerzo, calidad del producto, así como productividad. Los resultados aquí reportados sugieren una mejoría parcial con respecto a la precisión de las estimaciones.

OBJETIVO:

Conocer el ritmo de trabajo individual que consta en sacar la media y la desviación estándar para que cada alumno determine el tiempo que tarda en cada etapa del proceso de desarrollo software y así tenga una noción y pueda mejorar su desempeño.

COMPETENCIA A DESARROLLAR:

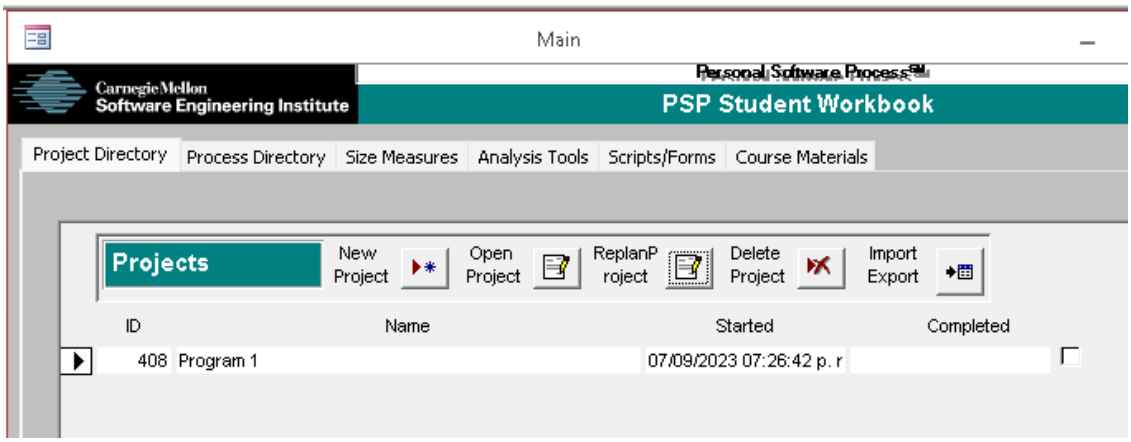
- Especificaciones de PSP y TSP.
- Que el alumno conozca su ritmo de trabajo y pueda hacer una evaluación del tiempo que tarda y con respecto a ello conozca su ritmo de trabajo en cada etapa.

MATERIAL Y EQUIPO (REQUERIMIENTOS):

- **Computadora personal.**
- **Dev C++**
- **Material de práctica de PSP0.**

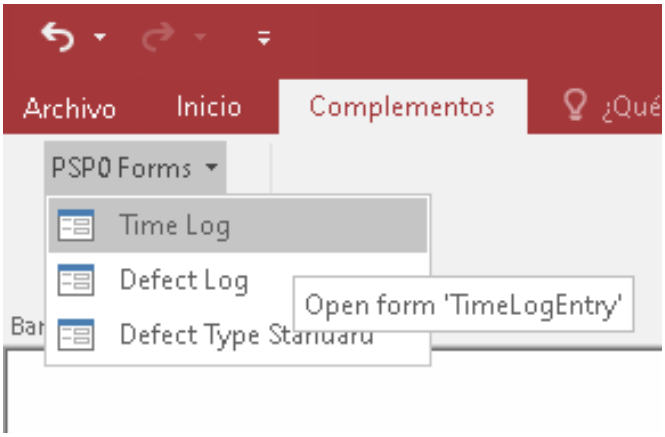
DESARROLLO:

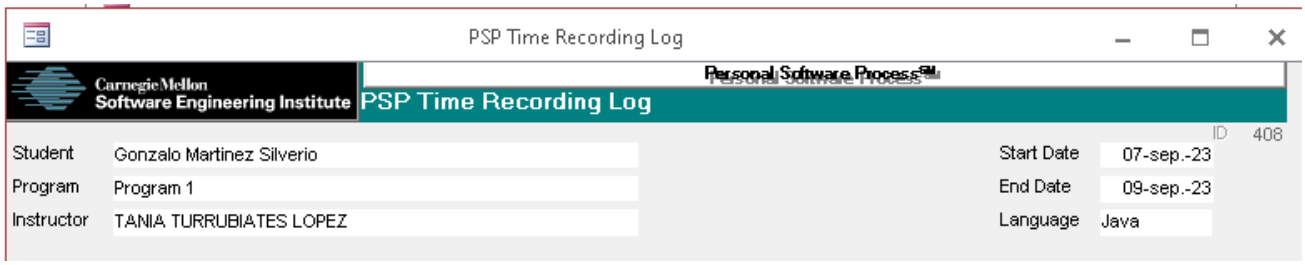
1. Registramos el inicio de la etapa de planeación dando doble clic en la barra blanca debajo de Started en la herramienta **PSP Student Workbook.mde**:



ETAPA DE PLANEACION:

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





3. El programa a realizar debe calcular la media y la desviación estándar de una serie de N números, para ello debe cumplir con las siguientes condiciones:

1.-Requerimientos del programa:

ID	PID	Phase	Start	Int.
16	408	PLAN	07/09/2023 07:27:42 p. r	0

- Los números a utilizarse deben ser capturados de forma externa ya sea con el teclado, archivo detexto, base de datos, etc.
- Los datos extraídos serán almacenados en un vector.

Los datos con los que trabajaremos son los siguientes:

Columna 1	Columna 2
Estimate Proxy Size	**Development Hours**
160	**15.0**
591	**69.9**
114	**6.5**
229	**22.4**
230	**28.4**
270	**65.9**
128	**19.4**
1657	**198.7**
624	**38.8**
1503	**138.2**

Utilizaremos las siguientes formulas:

Para la media:

$$x_{avg} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Para la desviación Estándar:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} \left(x_i - x_{avg}\right)^2}{n-1}}$$

2.-Estimación de tiempo:

El tiempo para este proyecto en todas sus etapas puesto que los requerimientos son bastante específicos será de 4 horas y 0 minutos.

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

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
16	408	PLAN	07/09/2023 07:27:42 p. r	0	07/09/2023 07:40:24 p. r	12.7	Requerimientos del sistema

ETAPA DE DISEÑO:

1.Registramos el inicio de la etapa de diseño en el Time Log
:

ID	PID	Phase	Start	Int.
16	408	PLAN	07/09/2023 07:27:42 p. r	0
17	408	DLD	07/09/2023 07:40:25 p. r	20

2.- El diseño nos presentará lo siguiente:

Este diagrama de clases muestra dos clases principales:

ProgramaPSP: Representa la clase principal que contiene el programa. Tiene dos **variables miembro privadas**: inicio y actual, que son punteros a nodos para mantener la lista enlazada. También contiene dos métodos públicos:
mediaYDesviacion(inicio: Nodo*, nVeces: int): Un método que calcula la media y la desviación estándar de la lista enlazada de números.
main(): El punto de entrada principal del programa.
Nodo: Representa la estructura para un nodo en la lista enlazada. Tiene dos **variables miembro privadas:** dato, que almacena un número, y siguiente, que es un puntero al siguiente nodo en la lista. En este diagrama de clases, las variables y métodos privados están indicados con un signo -, mientras que las variables y métodos públicos están indicados con un signo +.

```
+-----------------------------------+
|            ProgramaPSP            |
+-----------------------------------+
| - inicio: Nodo*                    |
| - actual: Nodo*                   |
| + mediaYDesviacion(inicio: Nodo*, nVeces: int): void
| + main(): int                     |
+-----------------------------------+


+--------------------------+
|            Nodo          |
+--------------------------+
| - dato: double           |
| - siguiente: Nodo*      |
+--------------------------+
```

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

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
16	408	PLAN	07/09/2023 07:27:42 p. r	0	07/09/2023 07:40:24 p. r	12.7	Requerimientos del sistema
17	408	DLD	07/09/2023 07:40:25 p. r	20	07/09/2023 08:59:33 p. r	59.1	Diseno y una Interrupcion para ir al baño

ETAPA DE CODIFICACION:

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

PSP Time

Carnegie Mellon Software Engineering Institute PSP Time Record

Student Gonzalo Martinez Silverio

Program Program 1

Instructor TANIA TURRUBIATES LOPEZ

ID	PID	Phase	Start	Int.
16	408	PLAN	07/09/2023 07:27:42 p. r	0
17	408	DLD	07/09/2023 07:40:25 p. r	20
18	408	CODE	07/09/2023 08:59:34 p. r	10

2.- Este código es creado lenguaje C, calcula la media y la desviación estándar de un conjunto de números ingresados por el usuario utilizando listas enlazadas. A continuación, se explica cómo funciona el código paso a paso:

Se incluyen las bibliotecas estándar necesarias, como stdio.h para entrada/salida estándar, stdlib.h para la gestión de memoria dinámica y math.h para operaciones matemáticas.

Se define una estructura llamada Nodo para representar un nodo en la lista enlazada. Cada nodo tiene un campo dato que almacena un número de tipo double y un puntero siguiente que apunta al siguiente nodo en la lista.

Se declara una función mediaYDesviacion que calcula la media y la desviación estándar de los números almacenados en la lista enlazada. Esta función toma dos argumentos: el inicio de la lista enlazada (inicio) y la cantidad de números en la lista (nVeces).

En la función main, se imprime información sobre el programa, como el nombre del instituto, el número de programa, el nombre del docente, el período escolar, el nombre del alumno y el grupo.

Se solicita al usuario que ingrese la cantidad de números que desea calcular (nVeces) mediante la función scanf.

Se inicializan dos punteros inicio y actual que se utilizarán para administrar la lista enlazada. Inicialmente, ambos punteros se establecen en NULL.

Se utiliza un bucle for para solicitar al usuario que ingrese los números uno por uno y almacenarlos en la lista enlazada. Cada número se agrega como un nuevo nodo en la lista enlazada.

Se utiliza la función malloc para asignar memoria dinámica para cada nuevo nodo en la lista enlazada y se almacena el número ingresado en el campo dato del nodo. Luego, se ajustan los punteros inicio y actual según sea necesario para mantener la estructura de la lista enlazada.

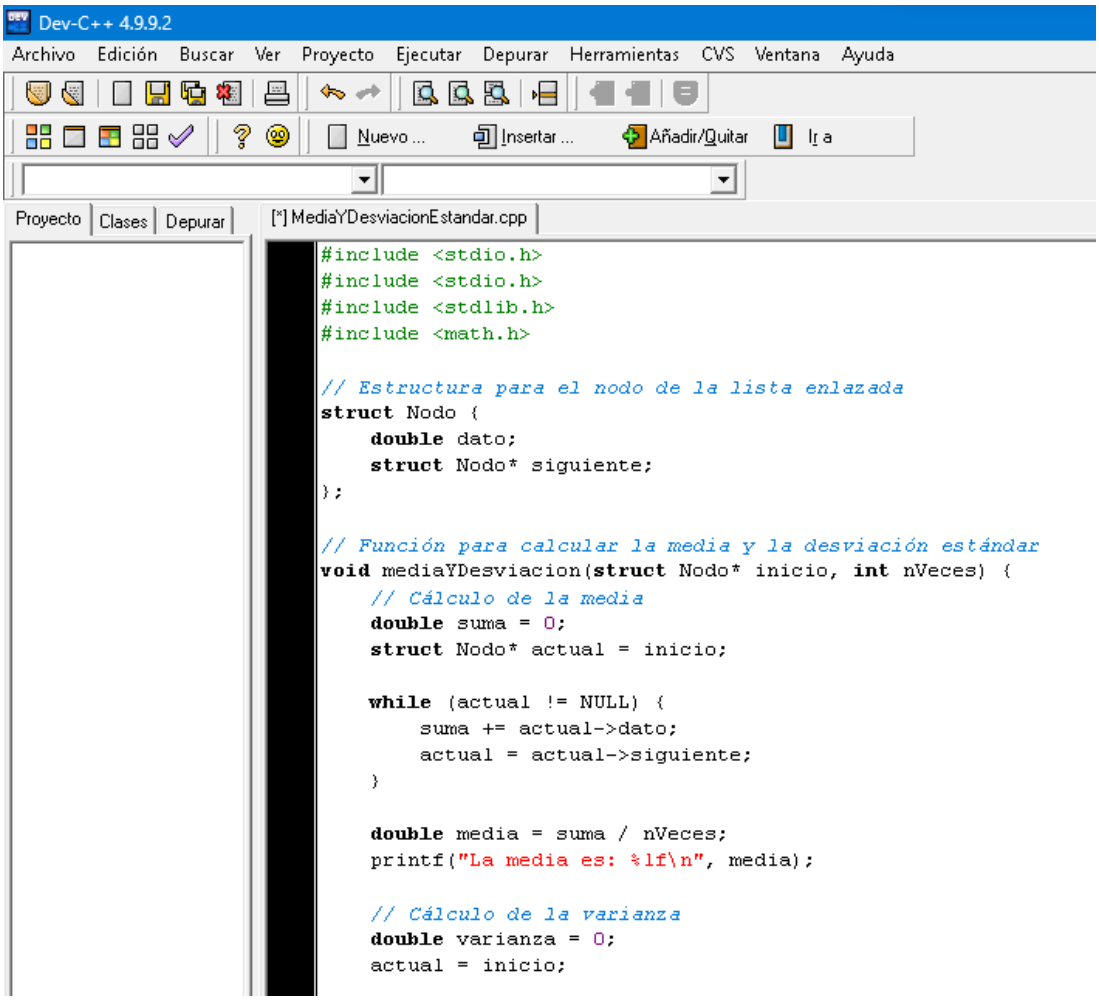
Después de almacenar todos los números, se llama a la función mediaYDesviacion pasando el inicio de la lista y la cantidad de números como argumentos.

La función mediaYDesviacion calcula la media y la desviación estándar utilizando los números almacenados en la lista enlazada. Para hacerlo, recorre la lista enlazada y realiza los cálculos necesarios. Luego, imprime los resultados en la consola.

Finalmente, en el main, se libera la memoria de los nodos de la lista enlazada utilizando free en un bucle while que recorre la lista. Esto se hace para evitar fugas de memoria.

El programa utiliza system("pause"); para pausar la ejecución en sistemas Windows antes de finalizar. Esto permite al usuario ver los resultados antes de que se cierre la ventana de la consola.

El programa termina con return 0 en la función main, indicando que se ejecutó con éxito.



```c
    while (actual != NULL) {
        varianza += pow(actual->dato - media, 2);
        actual = actual->siguiente;
    }

    varianza = varianza / (nVeces - 1);

    // Cálculo de la desviación estándar
    double desviacion = sqrt(varianza);
    printf("La desviacion estandar es: %lf\n", desviacion);
}

int main() {
    printf("INSTITUTO TECNOLOGICO SUPERIOR DE ALAMO TEMAPACHE\n");
    printf("PSP PROCESO PERSONAL DEL SOFTWARE || PROGRAMA: 01\n");
    printf("DOCENTE: DR. TANIA TURRUBIATES LOPEZ\n");
    printf("PERIODO ESCOLAR: AGOSTO 2023 - ENERO 2024\n");
    printf("ALUMNO: GONZALO MARTINEZ SILVERIO || 202Z0029\n");
    printf("||  GRUPO: 7S1A  ||\n");

    // Pedir al usuario cuántos números desea calcular
    int nVeces;
    printf("\n¿Cuantos numeros desea calcular? ");
    scanf("%d", &nVeces);

    struct Nodo* inicio = NULL;
    struct Nodo* actual = NULL;

    // Solicitar y almacenar los números en la lista enlazada
    for (int i = 0; i < nVeces; i++) {
        double numero;
        printf("Ingrese el numero %d: ", i + 1);
        scanf("%lf", &numero);
        // Crear un nuevo nodo para el número
        struct Nodo* nuevo = (struct Nodo*)malloc(sizeof(struct Nodo));
        nuevo->dato = numero;
        nuevo->siguiente = NULL;
        if (inicio == NULL) {
            inicio = nuevo;
            actual = inicio;
        } else {
            actual->siguiente = nuevo;
            actual = nuevo;
        }
    }
    // Calcular la media y la desviación estándar
    mediaYDesviacion(inicio, nVeces);
    // Liberar la memoria de los nodos de la lista enlazada
    actual = inicio;
    while (actual != NULL) {
        struct Nodo* temp = actual;
        actual = actual->siguiente;
        free(temp);
    }
system("pause");
    return 0;
}
```

Compilador | Recursos | Resultado de la compilación | Depurar | Ver Resultados

89: 3 | Modificac | Insertar | Líneas del Archivo: 104

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

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute — Personal Software Process℠

PSP Time Recording Log

ID 408

Student: Gonzalo Martinez Silverio
Program: Program 1
Instructor: TANIA TURRUBIATES LOPEZ
Start Date: 07-sep.-23
End Date: 09-sep.-23
Language: Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
16	408	PLAN	07/09/2023 07:27:42 p. r	0	07/09/2023 07:40:24 p. r	12.7	Requerimientos del sistema
17	408	DLD	07/09/2023 07:40:25 p. r	20	07/09/2023 08:59:33 p. r	59.1	Diseno y una Interrupcion para ir al baño
18	408	CODE	07/09/2023 08:59:34 p. r	10	07/09/2023 09:59:50 p. r	50.3	Codificacion y fui a beber agua

ETAPA DE COMPILACION:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute — Personal Software Process℠

PSP Time Recording Log

ID 408

Student: Gonzalo Martinez Silverio
Program: Program 1
Instructor: TANIA TURRUBIATES LOPEZ
Start Date: 07-sep.-23
End Date: 09-sep.-23
Language: Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
16	408	PLAN	07/09/2023 07:27:42 p. r	0	07/09/2023 07:40:24 p. r	12.7	Requerimientos del sistema
17	408	DLD	07/09/2023 07:40:25 p. r	20	07/09/2023 08:59:33 p. r	59.1	Diseno y una Interrupcion para ir al baño
18	408	CODE	07/09/2023 08:59:34 p. r	10	07/09/2023 09:59:50 p. r	50.3	Codificacion y fui a beber agua
25	408	PM	07/09/2023 09:59:51 p. r	2689	09/09/2023 07:34:05 p. r	45.2	Dormir, bañarme, Hice otras actividades, f
22	408	COMPILE	09/09/2023 09:59:51 p. r	0	09/09/2023 10:00:15 p. r	0.4	Compilamos el programa

Se compila:

```
BUILD SUCCESSFUL (total time: 4 seconds)
```

Compilación antes finalizar esta etapa se decidió hacer una prueba ejecutando el programa e introduciéndole datos y no se encontró algún error el cual procederé a solucionar en esta siguiente etapa de pruebas:

ETAPA DE TESTEO:

Registramos el inicio de nuestra etapa de Testeo:

PSP Time

Carnegie Mellon Software Engineering Institute — PSP Time Recor

Student: Gonzalo Martinez Silverio
Program: Program 1
Instructor: TANIA TURRUBIATES LOPEZ

ID	PID	Phase	Start	Int.
16	408	PLAN	07/09/2023 07:27:42 p. r	0
17	408	DLD	07/09/2023 07:40:25 p. r	20
18	408	CODE	07/09/2023 08:59:34 p. r	10
25	408	PM	07/09/2023 09:59:51 p. r	2689
22	408	COMPILE	09/09/2023 09:59:51 p. r	0
23	408	UT	09/09/2023 10:00:16 p. r	31

AHORA PROCEDEMOS A EJECUTAR EL PROGRAMA:

Introducimos los datos:

```
C:\Users\Gonzalo\Desktop\M
INSTITUTO TECNOLOGICO SUPERIOR DE ALAMO TEMAPACHE
PSP PROCESO PERSONAL DEL SOFTWARE || PROGRAMA: 01
DOCENTE: DR. TANIA TURRUBIATES LOPEZ
PERIODO ESCOLAR: AGOSTO 2023 - ENERO 2024
ALUMNO: GONZALO MARTINEZ SILVERIO || 202Z0029
||  GRUPO: 7S1A  ||

¬Cuantos numeros desea calcular? 10
Ingrese el numero 1: 160
Ingrese el numero 2: 591
Ingrese el numero 3: 114
Ingrese el numero 4: 229
Ingrese el numero 5: 230
Ingrese el numero 6: 270
Ingrese el numero 7: 128
Ingrese el numero 8: 1657
Ingrese el numero 9: 624
Ingrese el numero 10: 1503
```

Resultados:

```
La media es: 550.600000
La desviacion estandar es: 572.026845
Presione una tecla para continuar . . .
```

Probaremos introduciendo los datos de la columna 2:

```
C:\Users\Gonzalo\Desktop\M

INSTITUTO TECNOLOGICO SUPERIOR DE ALAMO TEMAPACHE
PSP PROCESO PERSONAL DEL SOFTWARE || PROGRAMA: 01
DOCENTE: DR. TANIA TURRUBIATES LOPEZ
PERIODO ESCOLAR: AGOSTO 2023 - ENERO 2024
ALUMNO: GONZALO MARTINEZ SILVERIO || 202Z0029
||  GRUPO: 7S1A  ||

¬Cuantos numeros desea calcular? 10
Ingrese el numero 1: 15.0
Ingrese el numero 2: 69.9
Ingrese el numero 3: 6.5
Ingrese el numero 4: 22.4
Ingrese el numero 5: 28.4
Ingrese el numero 6: 65.9
Ingrese el numero 7: 19.4
Ingrese el numero 8: 198.7
Ingrese el numero 9: 38.8
Ingrese el numero 10: 138.3
```

Resultados:

```
La media es: 60.330000
La desviacion estandar es: 62.269737
Presione una tecla para continuar . . .
```

Los resultados fueron verificados y son correctos, por lo tanto, marcare como finalizada la etapa de pruebas:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute — Personal Software Process℠

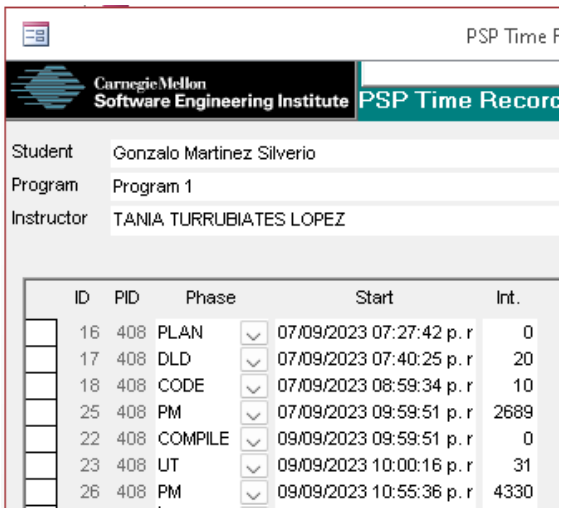
PSP Time Recording Log

ID 408

Student	Gonzalo Martinez Silverio	Start Date	07-sep.-23
Program	Program 1	End Date	09-sep.-23
Instructor	TANIA TURRUBIATES LOPEZ	Language	Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
16	408	PLAN	07/09/2023 07:27:42 p. r	0	07/09/2023 07:40:24 p. r	12.7	Requerimientos del sistema
17	408	DLD	07/09/2023 07:40:25 p. r	20	07/09/2023 08:59:33 p. r	59.1	Diseno y una Interrupcion para ir al baño
18	408	CODE	07/09/2023 08:59:34 p. r	10	07/09/2023 09:59:50 p. r	50.3	Codificacion y fui a beber agua
25	408	PM	07/09/2023 09:59:51 p. r	2689	09/09/2023 07:34:05 p. r	45.2	Dormir, bañarme, Hice otras actividades, f
22	408	COMPILE	09/09/2023 09:59:51 p. r	0	09/09/2023 10:00:15 p. r	0.4	Compilamos el programa
23	408	UT	09/09/2023 10:00:16 p. r	31	09/09/2023 10:55:34 p. r	24.3	Pruebas con el programa (cene)

Iniciamos la etapa del POSMORTEM



PSP Time F

Carnegie Mellon Software Engineering Institute PSP Time Record

Student Gonzalo Martinez Silverio

Program Program 1

Instructor TANIA TURRUBIATES LOPEZ

ID	PID	Phase	Start	Int.
16	408	PLAN	07/09/2023 07:27:42 p. r	0
17	408	DLD	07/09/2023 07:40:25 p. r	20
18	408	CODE	07/09/2023 08:59:34 p. r	10
25	408	PM	07/09/2023 09:59:51 p. r	2689
22	408	COMPILE	09/09/2023 09:59:51 p. r	0
23	408	UT	09/09/2023 10:00:16 p. r	31
26	408	PM	09/09/2023 10:55:36 p. r	4330

ETAPA POSMORTEM:

Haremos un análisis profundo de nuestro proceso de desarrollo que realizamos anteriormente enfocado en cada etapa, evaluando nuestro desempeño y los resultados obtenidos:

ETAPA DE PLANEACION:

Inicie con el análisis de la descripción del problema y comprendí lo que se nos pedía en los requerimientos para nuestro programa, en base a esto elabore una estimación del tiempo que tardare en realizar el programa la cual fue superada.

ETAPA DE DISEÑO:

Gracias a lo comprendido anteriormente procedí a elaborar el diseño en base al código realizado en la etapa anterior.

ETAPA DE CODIFICACION:

Todo se creó en una sola clase, pero al no haber un diseño detallado de esta por parte del programador, el tiempo de codificación aumento considerablemente puesto que se obtuvo algunos errores de sintaxis.

ETAPA DE COMPILACION:

Debido a la corrección de errores de sintaxis en la etapa anterior nuestro proceso finalizocon éxito y sin errores de sintaxis.

ETAPA DE PRUEBAS:

En esta etapa de pruebas solo se verifico si existía algún tipo de errores de los cuales no se encontró ninguno.

EVALUACION DE RESULTADOS:

METAS ESTABLECIDAS:

Metas principales en base a losrequerimientos:

1. El programa debía recibir los datos de manera externa (teclado, archivo de texto, base dedatos).
2. El programa debía implementar un arreglo para almacenar los datos.
3. El programa obtenga y muestre los resultadoscorrectos y precisos.

METAS NO CUMPLIDAS:

1. El tiempo estimado para la elaboración de este proyecto fue de 3hr 10 min.
2. Reducir aún más el tiempo en el desarrollo y aprender cómo funcionan en su totalidad las listas enlazadas.

METAS CUMPLIDAS:

En los datos: El programa logro exitosamente y sin mayor complicación leer los datos que se leintrodujeron, elaborar las listas enlazadas y la implementación de un ArrayList para el almacenamiento de datos.

FINAL DEL POSMORTEM

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute — Personal Software Process℠

PSP Time Recording Log

ID 408

Student	Gonzalo Martinez Silverio	Start Date	07-sep.-23
Program	Program 1	End Date	09-sep.-23
Instructor	TANIA TURRUBIATES LOPEZ	Language	Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
16	408	PLAN	07/09/2023 07:27:42 p. r	0	07/09/2023 07:40:24 p. r	12.7	Requerimientos del sistema
17	408	DLD	07/09/2023 07:40:25 p. r	20	07/09/2023 08:59:33 p. r	59.1	Diseno y una Interrupcion para ir al baño
18	408	CODE	07/09/2023 08:59:34 p. r	10	07/09/2023 09:59:50 p. r	50.3	Codificacion y fui a beber agua
25	408	PM	07/09/2023 09:59:51 p. r	2689	09/09/2023 07:34:05 p. r	45.2	Dormir, bañarme, Hice otras actividades, f
22	408	COMPILE	09/09/2023 09:59:51 p. r	0	09/09/2023 10:00:15 p. r	0.4	Compilamos el programa
23	408	UT	09/09/2023 10:00:16 p. r	31	09/09/2023 10:55:34 p. r	24.3	Pruebas con el programa (cene)
26	408	PM	09/09/2023 10:55:36 p. r	4330	12/09/2023 11:56:42 p. r	51.1	Postmortem
evo)	408			0		0.0	

RESULTADOS:

Los datos mostrados son confiables ya que el programa respeta cada operación de las fórmulas correspondientes.

RESULTADOS:

PSP Time Recording Log

Carnegie Mellon Software Engineering Institute — Personal Software Process℠

PSP Time Recording Log

ID 408

Student	Gonzalo Martinez Silverio	Start Date	07-sep.-23
Program	Program 1	End Date	09-sep.-23
Instructor	TANIA TURRUBIATES LOPEZ	Language	Java

ID	PID	Phase	Start	Int.	Stop Date and Time	Delta	Comments
16	408	PLAN	07/09/2023 07:27:42 p. r	0	07/09/2023 07:40:24 p. r	12.7	Requerimientos del sistema
17	408	DLD	07/09/2023 07:40:25 p. r	20	07/09/2023 08:59:33 p. r	59.1	Diseno y una Interrupcion para ir al baño
18	408	CODE	07/09/2023 08:59:34 p. r	10	07/09/2023 09:59:50 p. r	50.3	Codificacion y fui a beber agua
25	408	PM	07/09/2023 09:59:51 p. r	2689	09/09/2023 07:34:05 p. r	45.2	Dormir, bañarme, Hice otras actividades, f
22	408	COMPILE	09/09/2023 09:59:51 p. r	0	09/09/2023 10:00:15 p. r	0.4	Compilamos el programa
23	408	UT	09/09/2023 10:00:16 p. r	31	09/09/2023 10:55:34 p. r	24.3	Pruebas con el programa (cene)
26	408	PM	09/09/2023 10:55:36 p. r	4330	12/09/2023 11:56:42 p. r	51.1	Postmortem
evo)	408			0		0.0	

Total 243.1

Registro: 1 de 1 — Sin filtro — Buscar

PSP0 Project Plan Summary

Carnegie Mellon Software Engineering Institute

Personal Software Process℠

PSP0 Project Plan Summary

ID 408

Student	Gonzalo Martinez Silverio	Start Date	07-sep.-23
Program	Program 1	End Date	09-sep.-23
Instructor	TANIA TURRUBIATES LOPEZ	Language	Java

Time in Phase

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

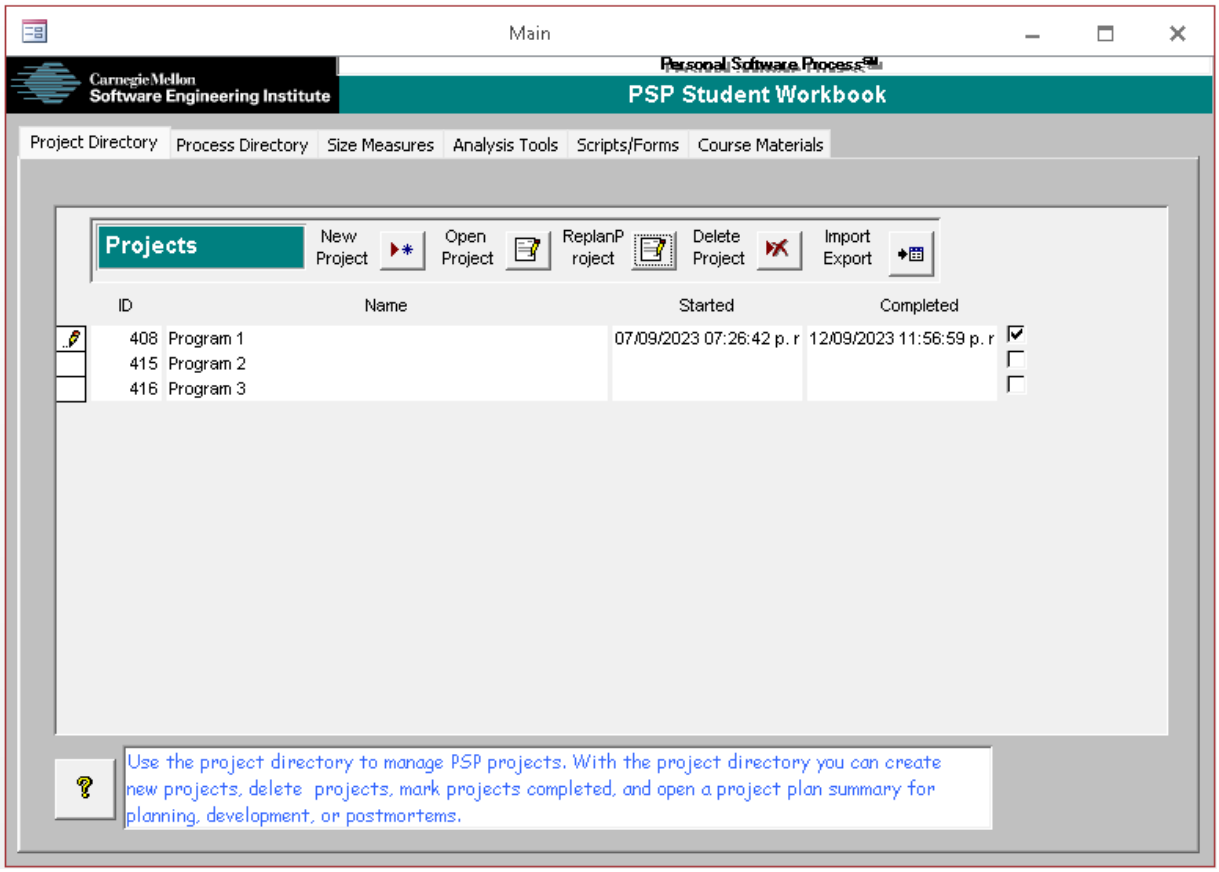
Defects Injected in Phase

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

Defects Removed in Phase

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

Registro: 1 de 1 Filtrado Buscar



CONCLUSIONES:

PSP (Personal Software Process) es un conjunto de prácticas y técnicas de mejora de procesos diseñadas para ayudar a los ingenieros de software a mejorar su productividad y la calidad de su trabajo. A continuación, se presenta una breve conclusión sobre el PSP:

Mejora de la Calidad: PSP se enfoca en mejorar la calidad del software al introducir una metodología disciplinada para el desarrollo de software. Esto implica la planificación, seguimiento y control rigurosos de todas las etapas del proceso de desarrollo.

Medición y Estimación: Una parte fundamental del PSP es la medición detallada de las actividades de desarrollo de software. Esto permite a los ingenieros recopilar datos objetivos sobre su propio rendimiento, lo que a su vez les ayuda a estimar de manera más precisa los tiempos y recursos necesarios para futuros proyectos.

Enfoque en la Calidad Personal: PSP se enfoca en el individuo y su habilidad para mejorar su propio desempeño. Ayuda a los ingenieros a identificar sus debilidades y áreas de mejora, lo que a su vez contribuye a un desarrollo más eficiente y de mayor calidad.

Proceso Escalable: PSP es escalable y puede adaptarse a diferentes tamaños de proyectos y organizaciones. Puede ser implementado tanto por ingenieros individuales como en equipos de desarrollo más grandes.

Reducción de Errores: Al utilizar prácticas de revisión y verificación, el PSP ayuda a reducir la cantidad de errores y defectos en el software, lo que ahorra tiempo y recursos a largo plazo al evitar correcciones costosas.

Es importante tener en cuenta que el proceso de software personal para el desarrollo de software ya que es uno de los elementos fundamentales para cumplir el desarrollo de software con calidad, ya que permite que los ingenieros que trabajan en proyectos de software puedan gestionar los tiempos de las tareas que se están realizando y controlando el tiempo eficiente y perdido mediante los formatos de control de tiempos y llevar el control de las faces en las cuales se tiene que terminar sin que contenga errores y si los tiene corregirlos rápidamente lo que permite a los desarrolladores tener más control sobre el proyecto que se está realizando, para obtener productos de calidad, el ingeniero necesita asumir su responsabilidad personal de la calidad de sus productos ya que los productos no se obtienen por azar, sino por el esfuerzo positivo para hacer un trabajo de calidad.

BIBLIOGRAFÍA (Formato APA)**:**

• 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