

LAB 1

18,30 1

DEPARTAMENTO:	DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN	CARRERA:	Telecomunicaciones		
ASIGNATURA:	FUNDAMENTOS DE PROGRAMACIÓN	PERIODO LECTIVO:	MAYO 2023 – SEPTIEMBRE 2023	NIVEL:	1ro
DOCENTE:	ING. OMAR QUIMBITA, MSc.	NRC:	8753	PRÁCTICA Nº:	1
TEMA DE LA PRÁCTICA:	IMPLEMENTACIÓN DE EJERCICIO CAJERO BANCO DEL PICHINCHA				

INTRODUCCIÓN:

PSeInt es un entorno de introducción a la lógica de programación que permite comprender los conceptos básicos de la programación. Es una herramienta gratuita y de código abierto que brinda una interfaz amigable para crear algoritmos y pseudocódigos. Sus aplicaciones en programación son diversas, desde la resolución de problemas simples hasta el diseño de algoritmos más complejos. PSeInt ofrece funciones como "según" y "mientras", que permiten tomar decisiones y repetir tareas respectivamente. Estas funciones son fundamentales para controlar el flujo del programa y solucionar problemas mediante estructuras condicionales y bucles. En resumen, PSeInt es una herramienta valiosa para aprender a programar y desarrollar habilidades algorítmicas.

Este laboratorio consistió en la creación de un pseudocódigo en Pseint para simular un banco o cajero Pichincha. El programa ofrecerá un menú de opciones que permitirá al usuario realizar diferentes acciones. Primero, podrá revisar o visualizar su saldo actual en la cuenta bancaria. Luego, tendrá la opción de retirar dinero de la cuenta. Por último, el usuario podrá finalizar el programa cuando lo desee. Este pseudocódigo permitirá a los usuarios interactuar con el sistema de manera intuitiva y realizar operaciones comunes en un banco o cajero, brindando una experiencia virtual similar a la de un cliente en un banco real.

OBJETIVOS:

- Crear una simulación de un cajero automático del Banco del Pichincha utilizando el lenguaje de programación PSeInt.
- Implementar la lógica necesaria para cada opción del menú, para llevar a cabo la operación correspondiente.
- Identificar y corregir cualquier error o comportamiento inesperado que se encuentre durante la programación del ejercicio.

MATERIALES:

EQUIPOS:

1. Computador
2. Programa pseint

INSTRUCCIONES:**ACTIVIDAD 1**

1. Investigar sobre estructura de control (1 Pto)

Las estructuras de control son el conjunto de reglas que permiten controlar el flujo de ejecución de las instrucciones

de un algoritmo

La mayoría de los lenguajes de programación actuales soportan o utilizan las mismas estructuras de control o al

Menos son muy similares. Lo que difiere de uno a otro es la sintaxis con la que están escritos y por tanto los

Compiladores o intérpretes los traducen a lenguaje máquina.

¿Para que sirve una estructura de control?

Las estructuras de control nos dan el poder de cambiar, controlar o modificar el orden o la secuencia en la que se

ejecutan las instrucciones del software a voluntad. Gracias a las estructuras de control, podemos abstraer

algoritmos o secuencias de comandos en una pieza de software para lograr su objetivo.

2. Descargar el instalador o ejecutable. (1 Pto)

<https://micampus.espe.edu.ec/mod/resource/view.php?id=62431>

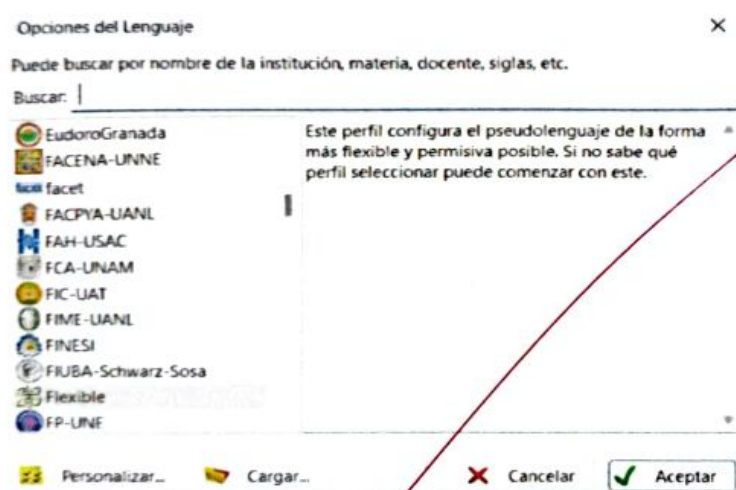


ACTIVIDAD 2

3. Instalar la aplicación (1 Pto)



4. Configurar la aplicación (1 Pto)



ACTIVIDAD 3

5. Implementar el ejercicio diagrama de flujo (2 Ptos) y pseint (12 Ptos)

```

("*****");
("    CAJERO AUTOMÁTICO BANCO PICHINCHA    ");
("*****");
("1. Consultar saldo");
("2. Retirar dinero");
("3. Salir");
("*****");
  
```



```

Escribir "Buenos dias"
Escribir "QUERIDO DE LAS PUERDAS APURAS ESPERAAAAA "
Escribir "Desayunas : Tostadas Pasa, Mielina Vascina, Galleta Juliana, Bryan Almeida"
Escribir ("AAAAAAAAAAAAAAAAAAAA")
Escribir ("CAJON AUTOMATICO BANCO PICHINCHA")
Escribir ("AAAAAAAAAAAAAAAAAAAA")
Escribir "Buenos dias como estas,
Definir monto como real, // Definimos la variable en este caso que sera monto
Escribir "1. Consultar saldo ", // Mostramos cada opcion para que nos presente en pantalla
Escribir "2. Retirar dinero ", // Mostramos cada opcion para que nos presente en pantalla
Escribir "3. Salir", // Mostramos cada opcion para que nos presente en pantalla
Escribir "Ingresa la opcion deseada ", // pedimos que seleccionen una de las 3 opciones
Leer monto, // Asigna la variable que para que se guarde el numero asignado
Finpor Fin Hacer // Mostramos al usuario para delatar que a terminado
1.
Escribir "Su saldo es de : 1000 dolares ", // Mostramos el saldo correspondiente a 1000 dolares para presentar en pantalla si el selecciona 1
2.
Escribir "Ingresa el monto a retirar ", // Para que seleccionen el monto que quieren retirar dinero
Leer monto,
Mientras monto > 0 o monto MOD 10 != 0 Hacer // Es decir si mientras para obtener multiples de 10 para asegurarse a lo mas de un cajero
Escribir "Ingresa un numero mayor a 0 y que sea multiplo de 10:", // pedir al usuario que ingrese un valor multiplo de 10 positivo
Leer monto, // Leer el monto que ingresen al usuario
Si monto > 0 o monto MOD 10 != 0 Entonces // Si ingresan condiciones para que el monto sea mayor a cero y sean multiples minimo multiples de 10
Escribir "El numero ingresado no cumple las condiciones. Inténtelo nuevamente.", // presentar en pantalla el mensaje si no cumple la condicion
Finsi
FinMientras
retiro = 1000 - monto, // proceso para obtener el saldo del retiro
Escribir "El monto correspondiente al retiro es : ", monto, // presentar en pantalla el monto que retiro el usuario
Escribir "Su saldo total es : ", retiro, // escribir el saldo despues de hacer el retiro en pantalla
Escribir "Salir ", // Selecciona si es para salir del programa
Otro Modo
Escribir "Opcion invalida intente de nuevo ", // en caso de seleccionar un numero fuera del rango escribir opcion invalida intente de nuevo finalizando el programa
FinHacer // Fin del programa

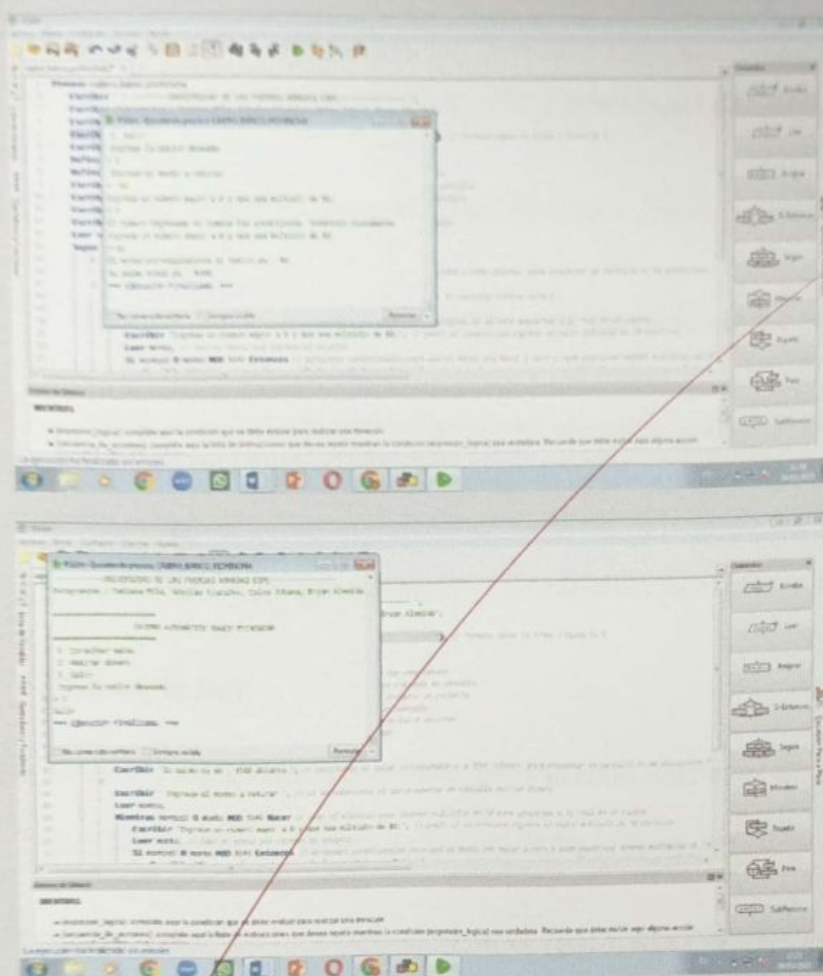
```

6. La cuenta debe tener un saldo de 1500 dólares (1 Pto)
7. Indicar con comentarios que realiza cada línea (1 Pto)

ACTIVIDADES POR DESARROLLAR EN EL LABORATORIO:

1. Implementación y/o modificación de las actividades propuestas
2. Analizar el ejercicio propuesto
3. Elegir el lenguaje de programación y la plataforma en la que se llevará a cabo el ejercicio.
4. Elaborar un menú de opciones similar al de un cajero automático
4. El cajero automático debe tener un proceso claro y fácil de usar
5. Definir la lógica detrás de las operaciones que el cajero puede realizar, como el retiro de dinero, la consulta de saldo, el depósito.
6. Valida y verifica las entradas
7. Considera todos los posibles errores que pueden ocurrir durante la ejecución del programa y proporciona mensajes de error claros y útiles
8. Antes de finalizar el ejercicio, realiza la ejecución que asegure de que el cajero automático funcione correctamente





RESULTADOS OBTENIDOS:

Como resultados obtenidos en la creación de un algoritmo en Pseint para simular el funcionamiento de un Banco Del Pichincha. Se desarrollaron tres funciones principales: consulta de saldo, retiro de dinero y opción de salida. Con este algoritmo, los usuarios podían verificar su saldo, retirar dinero y finalizar la sesión. El programa fue diseñado de forma amigable y fácil de usar, con una interfaz intuitiva. Se emplearon estructuras de control y variables para gestionar las transacciones bancarias. El proyecto resultó exitoso al proporcionar una solución sencilla y práctica para simular operaciones bancarias básicas.

CONCLUSIONES:

El Laboratorio que consistió en la creación de un algoritmo de simulación de un banco, fue un resultado altamente satisfactorio. Esta experiencia nos permitió afianzar los conocimientos teóricos adquiridos en clases al utilizar funciones de Pseint, asignación de variables y análisis de problemas. Los diagramas de flujo fueron una herramienta clave para visualizar el proceso y mejorar la lógica del algoritmo.

RECOMENDACIONES:

- Se recomienda conocer y entender el problema planteado al resolver un ejercicio de programación para así reconocer el punto de inicio para la solución del problema y tener claro el resultado que se desea obtener.
- Diseñar de forma cronológica el desarrollo de la solución identificando las variables y las constantes del problema.
- El diseño de la solución debe ser lo más general posible además se recomienda agregar comentarios de ser necesarios para la mejor comprensión del código.
- Simplificar los procesos desarrollados para obtener una solución más simple.
- Resolver el diagrama de flujo en lenguaje de pseudo-código y realizar pruebas de resultados con base a este diagrama.
- Escoger el lenguaje de programación más adecuado para traducir el pseudo-código.

REFERENCIAS

¿Qué es PSeInt? (n.d.). Fernandomonroytenorio.com. Retrieved May 31, 2023, from <https://fernandomonroytenorio.com/tema/que-es-pseint/>

RECOMENDACIONES PARA ALGORITMOS. (n.d.). RECOMENDACIONES PARA ALGORITMOS. Retrieved May 31, 2023, from <https://victorpedraza.wordpress.com/>

PSeInt. (n.d.). Sourceforge.net. Retrieved May 31, 2023, from <https://pseint.sourceforge.net/>

ELABORADO POR:
Bryan Almeida
Tatiana Pila
Johana Caiza
Nicolas Vizcaino