

Informe Académico: Análisis de Algoritmos Clásicos en PSeInt

1. Introducción al Pseudocódigo

El **pseudocódigo** constituye una herramienta fundamental en la formación de programadores, actuando como un puente lingüístico entre el pensamiento lógico humano y la sintaxis rigurosa de los lenguajes de programación. Su propósito principal es permitir la abstracción de la lógica algorítmica, facilitando el diseño de soluciones sin las restricciones inmediatas de un compilador. A través de PSeInt, los estudiantes pueden visualizar y ejecutar estas estructuras, consolidando su comprensión sobre el flujo de control y la manipulación de datos en un entorno controlado y didáctico.

2. Cuerpo del Informe: Análisis de Ejercicios

A. Algoritmo de Juego del Ahorcado

El algoritmo denominado **Ahorcado Corregido** gestiona una dinámica de adivinanza de palabras mediante el uso de estructuras de datos lineales.

Atributo	Descripción
Entradas	Caracteres individuales ingresados por el usuario para adivinar la palabra oculta.
Proceso	El sistema selecciona una palabra de un arreglo predefinido y utiliza un vector de progreso para comparar cada entrada. Valida la existencia de la letra y actualiza el estado del juego o reduce las vidas disponibles.
Salidas	Representación visual del progreso de la palabra y el estado final del jugador (Victoria/Derrota).

En cuanto a su arquitectura técnica, el código implementa **asignación de variables**, **condicionales complejos** para la lógica de aciertos, **bucles anidados** para el recorrido de cadenas y **arreglos** para el almacenamiento de las palabras y el progreso.

Prueba de Escritorio:

Palabra Objetivo	Letra Ingresada	¿Acierto?	Vidas Restantes
LOGICA	L	Sí	6
LOGICA	X	No	5

B. Sistema de Cajero Automático

Este algoritmo simula las operaciones financieras básicas de una entidad bancaria, priorizando la integridad del saldo del usuario.

Atributo	Descripción
Entradas	Clave numérica de seguridad, selección de menú y montos monetarios para transacciones.
Proceso	Tras una autenticación inicial, el programa entra en un ciclo de gestión donde suma o resta valores a la variable de saldo según la operación seleccionada.
Salidas	Despliegue del saldo actualizado y mensajes informativos sobre el éxito de las transacciones.

El análisis estructural revela el uso intensivo de **condicionales de selección múltiple (Segun)** y **bucles de repetición controlada**, omitiendo en este caso el uso de arreglos al manejar una única cuenta de usuario.

Prueba de Escritorio:

Saldo Inicial	Operación	Monto Transacción	Saldo Resultante
1000	Depósito	500	1500
1500	Retiro	200	1300

C. Simulación de Carrera de Autos

La **Carrera de Autos** es un ejemplo de simulación basada en eventos aleatorios y representación gráfica rudimentaria en consola.

Atributo	Descripción
Entradas	No requiere interacción directa del usuario durante la ejecución de la carrera.
Proceso	Ejecuta un ciclo donde dos variables de posición se incrementan mediante la función <code>Azar</code> , simulando el avance de dos vehículos hasta alcanzar un límite preestablecido.
Salidas	Animación en tiempo real de los vehículos y declaración del ganador en pantalla.

Este ejemplo destaca por su uso de **bucles de espera** y **lógica de azar**, integrando **asignaciones dinámicas** dentro de una estructura iterativa `Mientras` .

Prueba de Escritorio:

Posición Auto A	Posición Auto B	Avance A	Avance B	Resultado
38	37	3	1	Auto A Gana

D. Gestión de Inventario Empresarial

Este algoritmo demuestra la utilidad de los arreglos para el manejo de múltiples registros de datos relacionados.

Atributo	Descripción
Entradas	Identificador numérico del producto y volumen de unidades para la venta.
Proceso	Cruza la información de tres arreglos paralelos (Nombres, Stock, Precios) para validar la disponibilidad y calcular el coste total de la operación.
Salidas	Listado detallado de existencias y comprobantes de venta con importes calculados.

Estructuralmente, es el ejemplo más completo al integrar **arreglos multidimensionales conceptuales**, **bucles de recorrido (Para)** y **validaciones lógicas de stock**.

Prueba de Escritorio:

Producto Seleccionado	Stock Actual	Cantidad Venta	¿Venta Exitosa?
Cuaderno	50	5	Sí (Stock final 45)
Mochila	10	15	No (Stock insuficiente)

3. Desafío: Propuestas de Mejora

Se han seleccionado tres algoritmos para implementar optimizaciones que aumentan su robustez y funcionalidad:

- Seguridad en Cajero Automático:** Se ha implementado un contador de intentos fallidos. Al alcanzar el tercer error consecutivo en el ingreso de la clave, el sistema bloquea el acceso, simulando una medida de seguridad real. Asimismo, se estableció un saldo mínimo de permanencia.
- Interactividad en Recomendaciones:** El menú original se ha extendido para permitir que el usuario aporte sus propias recomendaciones a la base de datos temporal, transformando un sistema de solo lectura en uno participativo.
- Búsqueda en Inventario:** Se optimizó el gestor de inventario añadiendo una función de búsqueda por nombre. Esto elimina la dependencia del usuario de conocer los ID internos de los productos, mejorando significativamente la experiencia de uso.

4. Conclusiones

Tras el análisis exhaustivo, se determina que el algoritmo de **Ordenamiento mediante Burbuja** representa el mayor desafío conceptual. La complejidad radica en la sincronización de los bucles anidados y la gestión de la variable auxiliar para el intercambio de posiciones, lo cual exige un alto nivel de abstracción sobre cómo se reorganizan los datos en la memoria del programa.

5. Anexos

Se adjuntan los archivos .psc originales y las versiones mejoradas resultantes de este análisis para su revisión en el repositorio correspondiente.