



CONTROL DE LECTURA
PENSAMIENTO COMPUTACIONAL Y PROGRAMACIÓN

FECHA	NIVEL	PROFESOR	CORREO ELECTRÓNICO
06-10-2023	III° medio A-B IV medio A-B	Ricardo Ruiz Lavín	rruiz@inmacthno.cl

INDICADORES DE LOGRO
<ul style="list-style-type: none">- Comprenden los aspectos técnicos de la depuración a partir de un texto escrito.- Reconocer las funciones básicas de la depuración en computación.

NOMBRE DEL ESTUDIANTE:	CURSO:

EXIGENCIA	PUNTAJE IDEAL	PUNTAJE NOTA 4,0	PUNTAJE OBTENIDO	NOTA
60%	17 PUNTOS	10 PUNTOS		

Depuración de programas

Objetivo de Aprendizaje:

Comprender las técnicas básicas de depuración de programas para revisar de manera precisa el flujo de ejecución de un programa y el valor de las variables; en su caso, corregir posibles errores.

Introducción

La mayoría de usuarios de un computador o dispositivo inteligente está familiarizado con la ejecución de programas, de forma general cuando se quiere ejecutar un programa se lo indica al sistema haciendo doble clic sobre él, algunos usuarios más expertos ejecutan un programa al introducir comandos en una consola, pero si el programa se lo permite y puede abrir el código fuente en un editor de texto se encontrará con una serie de símbolos, números u otros caracteres que hace que el usuario se pregunte **¿Cómo y quiénes lograron escribir el código?**, es este caso son los ingenieros de software los responsables de escribir el código, previo proceso de análisis y diseño.



COLEGIO INMACULADA CONCEPCIÓN
TALCAHUANO
CICLO PADERBORN
DEPARTAMENTO DE MATEMÁTICA



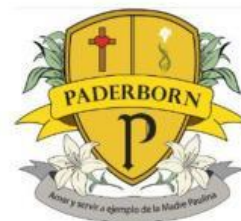
Un ingeniero en sistemas o de software identifica problemas y propone soluciones en diversas áreas del conocimiento, por tal motivo su trabajo va más allá de ser capaz de manipular diversas aplicaciones de ofimática, de navegar en internet, de reparar una computadora o de programar una computadora. Un ingeniero en sistemas crea y hace uso de diferentes niveles de abstracción, para entender y resolver problemas con mayor eficacia. La esencia de la programación es la abstracción y que las abstracciones para la informática son las herramientas intelectuales y la computadora una herramienta que automatizan las abstracciones.

Para un estudiante de secundaria le es bastante llamativa el área de la informática, las posibilidades de aprender a navegar en internet y utilizar todos sus servicios, de instalar programas, descargar juegos e incluso prácticas de hacking, todo esto le hace pensar que es una carrera muy interesante y nace su interés también por aprender a programar. Cuando un estudiante se interesa por aprender programación se enfrenta con el primer problema de no saber exactamente por dónde comenzar a estudiar. Como primeros requerimientos se destaca la dedicación en practicar y aprender de los errores, la afinidad con matemática e inglés. Una estrategia valedera es comenzar a enseñar programación utilizando los algoritmos como recursos esquemáticos para plasmar el modelo de la resolución de un problema.

El proceso de escribir los algoritmos hasta ponerlos a punto es operativamente complicado, más aún si se trabaja solo el lápiz y papel, tiza y pizarrón, o una herramienta de pseudolenguaje. El principal inconveniente es comprobar la validez del algoritmo, es difícil, mental o gráficamente, llevar a cabo las acciones del algoritmo en ejecución de manera totalmente objetiva sin dejarse llevar por la subjetividad de su especificación, es decir, tratando de olvidar el pensamiento que llevó al desarrollo y concentrándose exclusivamente en lo que se encuentra escrito.



COLEGIO INMACULADA CONCEPCIÓN
TALCAHUANO
CICLO PADERBORN
DEPARTAMENTO DE MATEMÁTICA



La palabra lógica viene del griego antiguo *logike* y significa razón, tratado o ciencia. En matemáticas es la ciencia que estudia los métodos de razonamiento, proporciona reglas y técnicas para determinar si un argumento es válido o no, indica la forma correcta de obtener conclusiones y los métodos adecuados para llegar a ellas. El razonamiento lógico se emplea en las matemáticas y la computación para demostrar teoremas y para resolver una multitud de situaciones problemáticas.

La lógica provee los elementos necesarios para representar el conocimiento a través de métodos de formalización de las frases declarativas. Cuando un sistema de razonamiento utiliza un lenguaje simbólico y un conjunto de reglas de inferencia recibe el nombre de lógica simbólica, esta lógica es la que a través de los algoritmos conformados por estructuras lógicas ha permitido el desarrollo de la informática, principalmente del componente de software.

Un algoritmo es un conjunto finito de reglas bien definidas en su lógica de control que permiten la solución de un problema en una cantidad finita de tiempo. La programación es parte esencial e integral de cualquier programa de ingeniería. Los estudiantes de ingeniería en sus últimos semestres tienen que enfrentar tareas de solución de problemas empleando esta competencia. El contar con buenas habilidades de desarrollo les ayudará a dar solución a estos problemas fácilmente. Es importante que los estudiantes de ingeniería y tecnología aprendan programación básica en sus primeros años de su preparación universitaria, principalmente la habilidad de depuración de programas. Depurar un programa significa someterlo a un ambiente de ejecución controlado por medio de herramientas dedicadas a ello. Este ambiente permite conocer exactamente el flujo de ejecución del programa, el valor que las variables adquieren, la pila de llamadas a funciones, entre otros aspectos. Es importante poder compilar el programa sin errores antes de depurarlo. Antes de continuar, es necesario conocer las siguientes definiciones (extraídas del Glosario IEEE610) ya que son parte latente del proceso de Desarrollo de Software:



COLEGIO INMACULADA CONCEPCIÓN
TALCAHUANO
CICLO PADERBORN
DEPARTAMENTO DE MATEMÁTICA



Error. Se refiere a una acción humana que produce o genera un resultado incorrecto.

Defecto (Fault). Es la manifestación de un error en el software. Un defecto es encontrado porque causa una Falla (failure).

Falla (failure). Es una desviación del servicio o resultado esperado.

La depuración de un programa es útil cuando:

- Se desea optimizar el programa: no basta que el programa se pueda compilar y se someta a pruebas que demuestren que funciona correctamente. Debe realizarse un análisis exhaustivo del mismo en ejecución para averiguar cuál es su flujo de operación y encontrar formas de mejorarlo (reducir el código, utilizar menos recursos llegando a los mismos resultados, hacer menos rebuscado al algoritmo), o bien, encontrar puntos donde puede fallar con ciertos tipos de entrada de datos.
- El programa tiene algún fallo: el programa no muestra los resultados que se esperan para cierta entrada de datos debido a que el programador cometió algún error durante el proceso de diseño. Muchas veces encontrar este tipo de fallos suele ser difícil, ya sea porque la percepción del programador no permite encontrar la falla en su diseño o porque la errata es muy pequeña, pero crucial. En este caso es de mucha utilidad conocer paso a paso cómo se ejecutan las estructuras de control, qué valor adquieren las variables, etc.
- El programa tiene un error de ejecución o defecto: cuando el programa está ejecutándose, éste se detiene inesperadamente. Suele ocurrir por error en el diseño o implementación del programa en las que no se contemplan las limitaciones del lenguaje de programación o el equipo donde el programa se ejecuta. Como el programa se detiene inesperadamente, no se conoce la parte del programa donde se provoca el defecto, teniendo que recurrir a la depuración para encontrarlo. El más común de este tipo de defecto es la “violación de segmento”.



Algunas *funciones básicas* que tienen en común la mayoría de los depuradores son las siguientes:

- Ejecutar el programa: se procede a ejecutar el programa en la herramienta de depuración ofreciendo diversas opciones para ello.
- Mostrar el código fuente del programa: muestra cuál fue el código fuente del programa con el número de línea con el fin de emular la ejecución del programa sobre éste, es decir, se indica qué parte del código fuente se está ejecutando a la hora de correr el programa.
- Punto de ruptura: también conocido por su traducción al inglés breakpoint, sirve para detener la ejecución del programa en algún punto indicado previamente por medio del número de línea. Como la ejecución del programa es más rápida de lo que podemos visualizar y entender, se suelen poner puntos de ruptura para conocer ciertos parámetros de la ejecución como el valor de las variables en determinados puntos del programa. También sirve para verificar hasta qué punto el programa se ejecuta sin problemas y en qué parte podría existir el error, esto es especialmente útil cuando existe un error de ejecución.
- Continuar: continúa con la ejecución del programa después del punto de ruptura.
- Ejecutar la siguiente instrucción: cuando la ejecución del programa se ha detenido por medio del depurador, esta función permite ejecutar una instrucción más y detener el programa de nuevo. Esto es útil cuando se desea estudiar detalladamente una pequeña sección del programa. Si en la ejecución existe una llamada a función se ingresará a ella.
- Ejecutar la siguiente línea: es muy similar a la función anterior, pero realizará todas las instrucciones necesarias hasta llegar a la siguiente línea de código. Si en la ejecución existe una llamada a función se ignorará.
- Ejecutar la instrucción o línea anterior: deshace el efecto provocado por alguna de las funciones anteriores para volver a repetir una sección del programa.
- Visualizar el valor de las variables: permite conocer el valor de alguna o varias variables.



Dependiendo de la herramienta usada para compilar el programa, si es de consola o determinial, su uso y las funciones disponibles variarán. En las IDE (Entornos de Desarrollo Interactivo), suelen existir herramientas de depuración integradas de manera gráfica. Es muy común que existan dos modos de desarrollar un programa y producir el archivo ejecutable que son "Debug" y "Release". El primer modo se recomienda exclusivamente durante el desarrollo del programa para poder depurarlo continuamente durante cualquier prueba de ejecución. El segundo modo se establece cuando el programa ha sido terminado y totalmente probado.

ÍTEM I: VERDADERO O FALSO. Coloca una V si la afirmación es verdadera, y una F si es falsa. (1 pto. c/u)

1. _____ Depurar un programa significa someterlo a un ambiente de ejecución no controlado por medio de herramientas dedicadas a ello.

2. _____ La depuración de un programa es útil solo cuando el programa tiene un error de ejecución o defecto.

3. _____ En las IDE (Entornos de Desarrollo Interactivo), suelen existir herramientas de depuración integradas de manera gráfica y simbólicas.

4. _____ El proceso de escribir los algoritmos hasta ponerlos a punto es operativamente complicado, necesitando solo lápiz y papel, tiza y pizarrón, o una herramienta de pseudolenguaje.

5. _____ Un ingeniero en sistemas o de software identifica problemas y propone soluciones en diversas áreas del conocimiento.



COLEGIO INMACULADA CONCEPCIÓN
TALCAHUANO
CICLO PADERBORN
DEPARTAMENTO DE MATEMÁTICA



ÍTEM II: Desarrollo: Responde claramente las siguientes preguntas. (2 pts. c/u)

1. Según el texto, ¿Por qué se sostiene que el proceso de escribir los algoritmos hasta ponerlos a punto es operativamente complicado?

2. ¿Qué son los puntos de ruptura?

3. ¿Cuáles son las diversas áreas del conocimiento que utiliza un ingeniero en sistemas o de software?

4. Describa con sus propias palabras en que consiste el concepto de lógica dado en el texto.

5. ¿Cuáles son algunas funciones básicas que tienen en común la mayoría de los depuradores (al menos 2)?

6. ¿Comparte usted lo indicado en el párrafo: "Para un estudiante de secundaria le es bastante llamativa el área de la informática..."? Comente basándose en su experiencia respecto a la asignatura electiva 2023.

"Adelante con valor y alegría"