



Universidad Nacional Autónoma de México



Facultad de Ingeniería

Ingeniería en computación (110)

Estructuras de Datos y Algoritmos I

Actividad asíncrona miércoles 01: Repaso de lo que aprendí
en la asignatura de Fundamentos de Programación.

Martínez Miranda Juan Carlos

(24/02/2021)

En el semestre 2021-1 cursé la materia de Fundamentos de Programación, en la cual aprendí diversos conceptos y realicé varios proyectos que mencionaré a continuación.

Durante la primera parte del curso se abordó el tema de algoritmos, donde aprendí que un algoritmo se trata de una serie de instrucciones para resolver un problema, un algoritmo debe presentar un orden, las instrucciones deben ser específicas, no debe presentar ambigüedades. Debe presentar entradas y salidas, antes del comienzo del algoritmo o durante la ejecución de este. Un algoritmo no puede tener una cantidad indefinida de pasos, debe terminar después de un número finito de pasos y tampoco debe ser demasiado complicado, pues se espera que su tiempo de ejecución sea muy tardado. En cada ocasión en la que se ejecute un algoritmo se debe obtener el mismo resultado, no deben existir variaciones. No debe llevar a un resultado erróneo y siempre debe llevar a un resultado en específico.

Para diseñar un algoritmo es importante saber qué es lo que deseamos obtener, el problema a solucionar y el resultado que debemos obtener, por lo cual se tiene que hacer un análisis del problema e identificar los posibles caminos a desarrollar, para esto, es importante identificar la solución que implique el menor tiempo y costos posibles.

De esta manera supe cómo realizar algoritmos correctamente para después, representarlos de una manera gráfica mediante el uso de diagramas de flujo, pues una vez teniendo el algoritmo en un diagrama de flujo, es más sencillo observar el orden de las instrucciones y cómo se deben llevar a cabo ciertos pasos dentro de este.

De la mano con los diagramas de flujo vimos el pseudocódigo, el cuál es la combinación de un lenguaje de programación y el idioma español en nuestro caso, se utiliza para la fácil comprensión de procesos e irse familiarizando con la estructura de los lenguajes de programación. Es recomendable el uso de pseudocódigo para la elaboración de algoritmos y que los diagramas de flujo sean más rápidos y sencillos de leer.

Una vez diseñados los diagramas de flujo y pseudocódigo, se deben realizar pruebas para determinar si se obtienen o no los resultados deseados, de esta manera podemos comprobar la utilidad del algoritmo diseñado y así realizar las correcciones necesarias o desecharlo totalmente y comenzar desde cero.

Gracias a esta parte del curso me familiaricé con la sintaxis de las funciones principales de programación, el uso de para 'x' con paso de 'n' hasta 'w', si 'x' hacer 'y', si no, hacer 'z', entre otros, pues de esta manera es más sencillo comprender lo que le estás pidiendo a la computadora que haga.

Para la segunda parte del curso la cual corresponde al lenguaje de programación C, utilizamos el entorno de desarrollo DevC++ para la elaboración de nuestros códigos. La complejidad al desarrollar el código del programa en este lenguaje es mayor en comparación con el pseudocódigo, pues se debe hacer uso de bibliotecas y una función principal en la cual se desarrolla todo el código del programa, al menos al principio del curso.

En C, existen diferentes tipos de datos, los básicos y cuyo uso es bastante común son:

Tipo de dato	Contenido
Int	Puede almacenar números enteros
Float	Puede almacenar números decimales
Char	Puede almacenar un carácter
String	Almacena una cadena de caracteres

Se hace uso de variables para almacenar datos, las variables son de un tipo determinado y se declaran de la siguiente manera: "int x,y,z;"

Las variables deben ser declaradas antes de ser utilizadas o se provocarán fallos en la ejecución del programa y simplemente no compilará, lo mismo pasa si se le asignan datos que no corresponden a un tipo de variable, no es posible asignarle datos de tipo carácter a variables de tipo entero ni viceversa, en algunos casos es posible que puedan almacenarse datos flotantes a datos enteros pero puede

ocasionar fallos en estos datos y por eso debe haber certeza de los datos que se le asignan a las variables.

Para guardar datos en variables previamente declaradas, se debe solicitar los datos por medio de una salida de texto al usuario y capturar la entrada ingresada por este mismo, la sintaxis que más se usó para capturar y almacenar los datos fue:

Tipo de dato	Sintaxis
Int	<code>scanf("%i",&x);</code>
Float	<code>scanf("%f",&y);</code>
Char	<code>scanf("%c",&z);</code>
String	<code>scanf("%s",palabra);</code>

Almacenar datos en variables es de mucha ayuda pues de esta manera podemos utilizar dichos valores para realizar tareas que los requieran dependiendo del objetivo del programa.

Para elaborar un programa podemos hacer uso de sentencias de control, estas nos permiten ejecutar instrucciones secuencialmente, repetir una serie de instrucciones mientras se cumpla una condición establecida o seleccionar que se ejecute una u otra instrucción dependiendo de una condición,

Un ejemplo de sentencia de control es la sentencia "if - else", cuya sintaxis es:

```
if (condición) {
```

```
(Serie de instrucciones a ejecutar en caso de que se cumpla la condición
```

```
} else {
```

```
(Instrucciones a ejecutar en caso de que no se cumpla la condición)
```

```
}
```

La sentencia “if – else” trabaja bajo una condición que nosotros establecemos, si se cumple la condición, realiza las series de instrucciones que nosotros definamos, en caso de no cumplirse la condición, se ignoran las líneas de código correspondientes al caso positivo y se ejecutan las que corresponden al caso contrario.

También existe la sentencia “switch” la cual funciona como un menú, su sintaxis es:

```
switch (opción) {  
    case 1:  
  
    case 2:  
  
    case 3:  
  
}
```

Es muy útil a la hora de preguntarle al usuario que elija una opción de las que le presentemos y así ejecutar exclusivamente esas instrucciones.

Una sentencia de control encargada de realizar la repetición de instrucciones es la sentencia “for”, cuya sintaxis es:

```
for (expresión 1; expresión 2; expresión 3) {  
  
    Instrucciones  
  
}
```

La sentencia “for” es de gran ayuda para llevar a cabo ciertas instrucciones un número definido de veces establecido por nosotros de manera controlada.

Estas sentencias son fundamentales para la elaboración de un programa, es muy importante cuándo y cómo saber usarlas para evitar redundancias en nuestro código, procurando hacerlo lo más eficaz posible.

Pude poner en práctica los conocimientos adquiridos durante la clase con la realización de un proyecto el cuál consistía en la elaboración de un juego que fue seleccionado por medio de una votación durante una clase, el juego que debíamos programar se llama “Deal or no deal”, este juego consiste en seleccionar un maletín

de un grupo de 26 maletines, dentro de cada uno de estos maletines se encontrará una cantidad de dinero, que puede ir desde \$1 hasta \$1,000,000, el maletín que el concursante seleccione será su recompensa si decide quedarse con él hasta el final del juego, a lo largo de las rondas se irán eliminando maletines, el concursante empieza eliminando 6 en la primera ronda abriendo uno por uno de este grupo de 6, este número disminuirá en uno con cada ronda que pase hasta llegar a un maletín, después de cada ronda recibirá una llamada del banco, este le ofrecerá una cantidad de dinero, el concursante puede elegir hacer un trato o no, si hace un trato con el banco tomará el dinero que se le ofreció y perderá lo que había dentro del maletín, si continúa seguirá eliminando maletines, las cantidades que revelan los maletines eliminados son premios que ya no se podrán obtener, por lo que si se rechaza la oferta del banco podría terminar con una cifra de dinero inferior a la que este ofrece.

Este proyecto fue realizado en equipo, yo fui el miembro encargado de realizar el código correspondiente al funcionamiento de registro de usuarios, inicio de sesión, estadísticas y el juego como tal, utilicé un menú principal que se encuentra dentro de la función principal y es el encargado de invocar a las funciones correspondientes a la selección del usuario mediante una estructura switch, en este se encuentran las opciones “Registro, inicio de sesión, estadísticas, salir”, dependiendo de qué escoja el usuario, cada opción llama a una función, si escoge la opción “salir”, el programa le despliega un mensaje de despedida y termina todos los procesos para cerrarse.

Para las funciones encargadas de guardar los datos de los usuarios y sus estadísticas hice uso de archivos de texto para poder tener acceso a dichos valores después de la ejecución del programa y puedan ser utilizados a largo plazo. Esta parte del código es la encargada de registrar a los usuarios para que puedan acceder al juego, cuando un usuario se registra se le solicita su usuario, se lee este usuario y se guarda para recorrer el archivo de principio a fin y comparar las cadenas de texto para verificar que no exista ningún usuario con ese mismo nombre, en caso de existir nombres de usuario iguales, se despliega un mensaje indicando que el usuario ya existe, y se le regresa a ingresar el usuario, de otra manera se procederá

a solicitarle su contraseña, una vez concluido esto se le notifica al usuario de que ha completado su registro mientras se imprime su usuario y contraseña en los documentos de texto correspondientes, posteriormente regresa al menú principal para seleccionar otra opción.

Para las estadísticas hice uso de tres funciones distintas, con el fin de cubrir los requerimientos del proyecto, la primera función se ocupa para leer el archivo de estadísticas y mostrar el nombre de usuario, cantidad de juegos jugados y la cifra total de dinero ganado de todos los jugadores existentes en dicho archivo, esta función es invocada desde el menú principal. La segunda es la encargada de mostrar el nombre de usuario, cantidad de juegos jugados y la cifra total de dinero ganado únicamente de un jugador, esta función es invocada dentro de un menú secundario al que accede el usuario al iniciar sesión. Para poder llamar a esta función es necesario iniciar sesión con el usuario del que se desean conocer las estadísticas. La tercera función es la más importante pues las demás solamente buscan e imprimen los datos que se encuentren en el archivo de estadísticas, esta tercera función tiene como tarea hacer el registro de los datos como nombre de usuario, cantidad de juegos jugados y dinero ganado. En esta función se hace la suma de la cantidad de dinero ganado, de los juegos jugados y se reescribe en otro archivo el cual reemplazará al original, se elimina el anterior, se renombra al nuevo, de manera que se mantengan actualizadas las estadísticas de todos los jugadores

Para el juego en sí, también utilicé una función, en esta, con un ciclo "for" se asignan aleatoriamente las cantidades a los maletines para que cada juego sea distinto. Después el usuario escoge su maletín y lo separamos del resto hasta que decida cambiarlo por la oferta del banco o si lo abre hasta el final de las rondas. Se abren los maletines restantes para revelar su contenido y el banco realiza una oferta con base en las cantidades que aún son probables que tenga el usuario, la oferta consiste en un promedio de las cantidades disponibles entre el número de maletines cerrados multiplicado por 0.15 de probabilidad de que el usuario tenga una cantidad considerable en su maletín, la probabilidad aumenta 0.05 en cada ronda, de esta

manera las ofertas se mantienen balanceadas y no se le dan pistas al usuario de la cantidad que pueda contener su maletín.

Ya sea que el usuario acepte la oferta del banco o prefiera guardar su maletín hasta el final, se asigna el valor a una variable que devolverá la función del juego y que la tercera función de estadísticas recibirá y registrará como se le instruyó.

La realización de este proyecto fue de gran ayuda para perfeccionar mis conocimientos de programación y reforzar algunos conceptos e incluso aprender nuevas maneras de realizar ciertos procesos, realmente considero que mi nivel de conocimiento acerca de programación es bueno y mejorará con el tiempo.