

Unidad 1: Introducción a la programación y los algoritmos

Introducción y orientaciones para el estudio

En el desarrollo de este módulo abordaremos:

- Definiciones fundamentales para entender qué es la programación.
- Resolución de problemas utilizando pseudocódigo (lenguaje natural).
- Uso de la toma de decisión mediante estructuras condicionales.
- Primer acercamiento a las estructuras repetitivas.

Objetivos

Pretendemos que al finalizar de estudiar esta unidad el alumno logre:

- Comprender qué es la programación.
- Realizar el análisis de problemas sencillos.
- Generar estrategias de resolución básicas.
- Utilizar pseudocódigo.

Aclaraciones previas al estudio

En este módulo, el alumnos encontrará:

- Contenidos
- Conceptualizaciones centrales
- Bibliografía

Usted debe tener presente que los contenidos presentados en el módulo no ahondan profundamente en el tema, sino que pretenden ser un recurso motivador para que, a través de la lectura del material, la bibliografía sugerida y el desarrollo de las actividades propuestas alcance los objetivos planteados.

Cada módulo constituye una guía cuya finalidad es facilitar su aprendizaje.

Introducción a la programación y los algoritmos

Dentro de los diferentes roles que pueden encontrarse relacionados con el área de sistemas, el de programador junto con el de analista, es uno de los más conocidos pero, ¿qué hace exactamente un programador?

El programador es el encargado de implementar un programa -valga la redundancia- en algún lenguaje de computación.

Esta es la tarea que desarrollarán en la asignatura de Taller de Computación I. Sin embargo, ningún programador realiza esta actividad sin seguir instrucciones suficientemente precisas, que obtiene de los algoritmos.

Un algoritmo es un conjunto finito (de algunas) de instrucciones, que llevan a la solución de un problema.

A diferencia de los programas, los algoritmos son expresados mediante pseudocódigo o diferentes tipos de diagrama, que iremos aprendiendo a medida que avancemos en la materia.

Esta asignatura intentará:

- crear en ustedes interés en la resolución de problemas,
- incentivar e incrementar su capacidad de abstracción,
- generar soluciones mediante diferentes modelos y métodos para obtener las herramientas para diagramar las soluciones mediante las instrucciones que un programador .

A partir de aquí comienzan a dar sus primeros pasos en la elaboración de programas.

Como su nombre lo indica, la programación, comprende la actividad de generar un plan de ejecución preestablecido para lograr un objetivo determinado. Noten que el concepto de programación es de uso cotidiano en diferentes medios, aunque en este espacio lo relacionamos directamente con la computación.

Algoritmos - Introducción

Como ya mencionamos, un algoritmo es un conjunto ordenado de instrucciones que, siguiendo un método y/o modelo planteado por un paradigma¹, sirve para llevar a cabo la resolución de un problema.

Aunque no lo sepamos, en la vida cotidiana estamos ejecutando algoritmos continuamente. Es decir, seguimos un conjunto de instrucciones implícitas que alguna vez aprendimos, y lo hacemos prácticamente de modo automático. Por ejemplo, cada vez que cocinamos algún plato de pasta, sabemos que debemos seguir una cantidad de instrucciones que ya conocemos, pero que pertenecen a una receta.

Veamos, primero debemos poner el agua a hervir, luego le agregamos sal y, si queremos, un poco de aceite. Cuando hierve, agregamos los fideos. Esperamos a que pasen 10

¹ Un paradigma de programación es un modelo básico de diseño e implementación de programas, que permite desarrollar programas conforme a ciertos principios o fundamentos específicos que se aceptan como válidos. En el sitio de la Universidad de Murcia (<http://www.um.es/docencia/barzana/IAGP/lagp3.html#BM3>) podemos leer "Un paradigma de programación es un modelo básico de diseño y desarrollo de programas, que permite producir programas con unas directrices específicas. (...) Un paradigma de programación es una colección de modelos conceptuales que juntos modelan el proceso de diseño y determinan, al final, la estructura de un programa".

minutos, apagamos la hornalla, ponemos manteca en el plato, colamos los fideos y los servimos.

Recordemos que en este caso, el problema era cocinar los fideos. Teniendo en cuenta este ejemplo, podemos ver que si la receta está escrita en español, inglés o italiano, no es importante, ya que el algoritmo plantea la solución del problema sin importar en que lenguaje o idioma esté escrito.

Basta que quien deba ejecutarlo, entienda ese lenguaje, para poder ejecutar las acciones indicadas.

Etapas en la resolución de un problema

La resolución de un problema en el área de sistemas consta de cuatro etapas, dos de las cuales abarcaremos en esta asignatura y dos en Taller de Computación I.

Las etapas son:

- Enunciación del problema: Formular el problema de forma clara, completa y concisa, evitando ambigüedades.
- Diagramar la solución: Analizar el problema, hallar la solución y diagramar el conjunto de instrucciones necesarias.
- Codificación: Transcribir las instrucciones generadas en los diagramas lógicos, utilizando algún lenguaje de computación.
- Ejecución: Llevar adelante la solución planteada al generar el diagrama e implementada en la etapa de codificación.

La ejecución puede llevar a la resolución correcta del problema, o bien mostrarnos un error. En este caso, debemos volver a las etapas anteriores, para encontrar si el error se produjo en la etapa de análisis o de codificación y corregirlo. Este proceso puede realizarse varias veces hasta depurar el programa.

Pseudocódigo

El pseudocódigo es una representación de instrucciones expresadas en lenguaje natural. Tienen un orden específico que nos marca el rumbo para resolver nuestro problema.

La realización de un pseudocódigo, permite varias cosas:

- utilizar un lenguaje común a todos los programadores;
- lograr un nivel de abstracción cuando se realizan programas;
- facilitar la traducción de las instrucciones a un lenguaje de programación.

En esta primera unidad, hablar de diagramar la solución implica expresar el algoritmo en pseudocódigo, proporcionando el conjunto de instrucciones, en el orden correcto, que llevará a la resolución del problema una vez ejecutadas.

Hasta ahora hemos trabajado con un conjunto de definiciones relacionadas con la programación. Ahora comenzaremos a profundizar cada tema con el objeto que cada uno pueda llegar a escribir sus propios programas.

Ejemplos

Ejemplo 1

Primer ejemplo de resolución de un problema a través de pseudocódigo

Para comenzar a tratar los conceptos de resolución de problemas, utilizaremos una situación que se puede dar en lo cotidiano.

Enunciado del Problema

Debo entrar a mi casa, en donde vivo solo, y acabo de notar que no tengo las llaves. Por suerte, mi vecina tiene un juego.

Lo primero que me debo preguntar es si “alcanza con el enunciado para vislumbrar la solución”.

Para plantear la solución de un problema primero debo analizar el enunciado del problema, y en caso de hacer falta, puedo mencionar ciertos puntos que aclaren los datos que creo pueden faltar. De esta manera “completo” el enunciado para poder encarar la propuesta de una solución.

Entonces, se definen hipótesis cuando los datos del enunciado no son suficientes para la resolución del problema.

Importante: nunca deben contradecir el enunciado

Para este problema definimos:

Hipótesis

Mi vecina del 9° B habitualmente está en su casa.

Datos

¿Qué datos tengo? De la lectura del enunciado surge que:

Mi vecina del 9° B tiene una copia de mis llaves.

Pasos de la solución

- Comunicarme con mi vecina.
- Explicarle la situación.
- Pedirle que me alcance mis llaves.

- Aguardar que me entregue las llaves.
- Agradecerle mucho.
- Ingresar a mi casa.

Se trata de una enumeración detallada y ordenada de los pasos a seguir para resolver un problema.

Hemos expresado las líneas de cada paso con un número pues las mismas expresan una secuencia. Esto quiere decir que el orden de la ejecución es línea a línea de acuerdo a los números.

Con este primer ejemplo, aprendimos a detallar paso a paso un problema (como en el caso de la receta de cocina) y que al finalizar, este conjunto de instrucciones, he solucionado mi problema.

La ejecución del problema desarrollado me va a permitir ingresar a mi casa.

Es importante también que la información proporcionada en el enunciado del problema cumpla con la regla de las “Tres Ce” (Claro, Conciso y Completo).

Incluso el comentario “en dónde vivo solo” es una aclaración implícita de que no hay nadie en mi casa que pueda abrirme la puerta.

En este primer problema aprendimos:

- cómo transformar un enunciado de un problema en un conjunto de pasos para resolver el mismo
- que la solución de un problema está compuesta por una secuencia de pasos

En el siguiente ejemplo introducimos lo que llamamos toma de decisión, esto es poder decidir entre varias alternativas en un momento dado. Por ejemplo decidimos salir con paraguas si está lloviendo o ir al cine si la película que pasan nos interesa.

Ejemplo 2

Toma de decisión. Segundo ejemplo de resolución de un problema a través de pseudocódigo.

Vamos a modificar el enunciado del ejemplo anterior para introducir la toma de decisión.

Enunciado del Problema

Debo entrar a mi casa, en donde vivo solo, y acabo de notar que no tengo las llaves. Por suerte mi vecina tiene un juego de llaves. Además, muchas veces llega a su casa pasadas las 22:30 Hs.

Hipótesis

Mi vecina del 9° B, siempre duerme en su casa.

Datos

Puede ser que no sea aún las 22 Hs.

Mi vecina llega muchas veces a su casa pasadas las 22:30 Hs.

Para resolver este problema nos surge un problema: puede que mi vecina no esté en su casa a la hora que llegamos.

Agregaremos entonces el primer concepto de toma de decisión: El “si” condicional, prevé que una situación planteada pueda no cumplirse, presentando una solución alternativa.

Análisis y estrategia de resolución

Se trata de analizar el problema (como en el ejemplo anterior) y elaborar una estrategia de solución del problema.

Según el planteo de este nuevo problema, es posible que mi vecina no se encuentre en su casa, de manera tal que debo prever la situación.

Pueden pasar dos cosas: que mi vecina esté o no esté. En cada caso deberé hacer algo diferente. Es posible que en cada caso deba hacer más de una actividad.

Al resolver mediante un diagrama en pseudocódigo este problema, veremos como una instrucción puede ser expandida a instrucciones más pequeñas y concretas, quitándole ambigüedad y generando una interpretación más certera.

En este caso debemos verificar si la vecina nos atiende el timbre, y en caso de no hacerlo, por ejemplo, ir a tomar un café.

Pasos de la solución

1. Comunicarme con mi vecina
 - 1.1. Tocar el timbre a mi vecina
 - 1.2. Si (mi vecina atendió el timbre)
 - 1.2.1. Entonces
 - 1.2.1.1. Pedirle que baje a abrirme
 - 1.2.1.2. Explicarle la situación
 - 1.2.1.3. Pedirle que me alcance mis llaves
 - 1.2.1.4. Aguardar que me entregue las llaves
 - 1.2.1.5. Agradecerle mucho
 - 1.2.1.6. Ingresar a mi casa
 - 1.2.2. Si no atendió el timbre
 - 1.2.2.1. Ir a tomar un café

La instrucción 1.2 y todas las sub-instrucciones que la componen, forman lo que se llama un ***si condicional***.

Esta estructura nos sirve para generar distintas resoluciones para una situación particular.

Mi vecina atendió el timbre es lo que llamaremos condición lógica o proposición. Una condición lógica está dada por un conjunto de una o más proposiciones lógicas, (en este caso, una sola), y solo puede devolver dos resultados: verdadero o falso. Si se fijan bien, la frase está enunciada de manera tal que puede ser verdadera o falsa.

Si la condición es verdadera (Mi vecina atendió el timbre), entonces se llevarán a cabo las subinstrucciones 1.2.1.1 a 1.2.1.6 (en general todas las indicadas como 1.2.1.x).

Si la condición es falsa, (si no), se llevará a cabo la instrucción 1.2.2.1.

Observemos que se utiliza una tabulación para indicar un incremento de nivel, esto permite establecer jerarquías en el pseudocódigo y permite identificar fácilmente el orden.

Sabemos que la instrucción 1 está compuesta por las instrucciones 1.1, 1.2 y 1.3, generalizándolo como 1.x donde x es cualquier número entero que marca el orden de la instrucción.

A su vez, las instrucciones 1.2 y 1.3 pueden a su vez estar compuestas por otro conjunto de instrucciones; para indicarlo, agregamos un número más en la numeración y tabulamos nuevamente. Cada vez que tabulamos, decimos que estamos indentando².

La situación que se planteó en este ejemplo (“toma de decisión”) se puede dar muchas veces en el programa. Incluso se puede dar de forma anidada.

Veamos con una pequeña modificación al ejemplo:

Mi vecina puede no estar antes de las diez y media de la noche y además la debo llamar al celular o la casa de su hija.

En este caso debo determinar si es más temprano de esa hora, y en caso de ser así debo llamar primero a la casa de la hija y si no atiende allí debo llamar a su celular. En este caso son procesos de selección anidados.

Otro ejemplo: debo llamar a mi vecina y puede ser que mi teléfono celular se haya quedado sin batería. En este caso debo verificar esto e ir a buscar un teléfono público si no tengo carga. Se trata de varios procesos de selección en un mismo problema.

² La indentación se realiza insertando tabulaciones a la izquierda (se corre el texto hacia la derecha), generando grupos lógicos que permitirán hacer un análisis más simple y encontrar posibles errores.

En el siguiente ejemplo introducimos un nuevo concepto: los procesos repetitivos. Se trata de repetir más de una vez un conjunto de pasos.

Ejemplo 3

Ciclo Mientras (condición) hacer – fin-Mientras

Continuando con el ejemplo anterior -y como último ejemplo para cerrar la idea de algoritmos y pseudocódigo- vamos a suponer que comunicarme con mi vecina puede costarme más trabajo, con lo cual tendré que probar varias veces hasta contactarme con ella. Podría, por ejemplo, tocarle el timbre y llamarla al celular hasta lograr la comunicación.

Enunciado del Problema

Deseo entrar a mi casa, en donde vivo solo, y acabo de notar que no tengo las llaves. Por suerte mi vecina tiene una copia de mis llaves. Además, muchas veces llega a su casa pasadas las 22:30 Hs. Cuento también con el número de su celular.

Hipótesis

*Mi vecina del 9° B, tiene una copia de mis llaves
Su celular siempre está prendido.*

Datos

*No son aún las 22 hs y mi vecina llega muchas veces a su casa pasadas las 22:30 hs.
Cuento con su número de teléfono celular.*

Análisis y estrategia de resolución

Como en el análisis del ejemplo anterior, vemos que mi vecina puede estar o no en su casa, pero ahora cuento, según los datos del problema, con su número de teléfono celular, lo cual me permitiría encontrarla en caso de que no estuviera.

La nueva estrategia es, tocar el timbre y en caso de que no me atienda, en lugar de ir a tomarme un café, llamarla hasta que me atienda el celular.

Veamos entonces el diagrama al cual llegaríamos aplicando la estrategia planteada.

Pasos de la solución

1. Comunicarme con mi vecina
 - 1.1. Tocar el timbre a mi vecina
 - 1.2. Si (mi vecina atendió el timbre)
 - 1.2.1. Entonces
 - 1.2.1.1. Pedirle que baje a abrirme
 - 1.2.2. si no atendió el timbre
 - 1.2.2.1. escribir el número de mi vecina en el celular

- 1.2.2.2. Llamar a mi vecina
- 1.2.2.3. Mientras (mi vecina no respondió) Hacer:
 - 1.2.2.3.1. escribir el número de mi vecina en el celular
 - 1.2.2.3.2. Llamar a mi vecina.
- 1.2.2.4. Fin-Mientras (vuelve a la instrucción 1.2.4)
- 1.2.2.5. Explicarle la situación
- 2. Pedirle que me alcance mis llaves
- 3. Aguardar que me entregue las llaves
- 4. Agradecerle mucho
- 5. Ingresar a mi casa

La instrucción 1.2.2.1 indica que las próximas subinstrucciones van a repetirse mientras que la condición mi vecina no respondió, se evalúe como verdadero. La instrucción 1.2.2.4 indica el fin de las subinstrucciones. Esta instrucción marca el retorno al origen del ciclo, el cual será evaluado nuevamente hasta que el resultado de la condición sea falso. En ese momento, continua el flujo del programa con la siguiente instrucción al fin-mientras.

Las instrucciones 1.2.2.1 y 1.2.2.2 son lo que se llama una lectura adelantada. Antes de comenzar el ciclo, hacemos una primera evaluación, cuyo resultado puede dar falso (que mi vecina atienda) y por lo tanto no precisar hacer un ciclo.

En el ejemplo:

- si mi vecina me responde la primera vez que la llamo, no preciso intentarlo nuevamente.
- si mi vecina no responde, entonces debo intentarlo varias veces, por lo tanto se ingresa en la estructura repetitiva.

Debemos leer con cuidado el ejemplo para poder comprender cómo vuelvo a ejecutar las líneas de más arriba.

Notemos que entre el “mientras” y el “fin-mientras” hay un conjunto de líneas que serán ejecutadas tantas veces (posiblemente, más de una) como sea necesaria, hasta que se cumpla la condición o situación que esperamos y que está asociada al “mientras”.

Siguiendo con el mismo ejemplo, cuando recibimos una llamada telefónica que no queremos atender, el contestador tarda algunos rings hasta que atiende automático. Esto pasa pues tiene programado un ciclo mientras fin-mientras, hasta que la cantidad de rings sea seis.

En síntesis

Ya disponemos de tres herramientas para expresar las estrategias para resolver problemas:

- la secuencia: permite ejecutar paso a paso en forma ordenada un conjunto de instrucciones;

- el si condicional: permite elegir diferentes cursos de acción, según se cumpla o no una condición;
- el ciclo mientras: permite llevar a cabo un conjunto de acciones mientras se cumple una condición.

También contamos con la posibilidad de generar aclaraciones sobre el enunciado, que le quiten la ambigüedad y aclaren la solución. A esta declaración la llamamos hipótesis, y no debe contradecir el enunciado, sólo aclararlo.

Analizar un problema es leer detenidamente el enunciado del mismo tratando de identificar los datos, los objetivos y si falta información para encarar la solución.

Es importante que antes de comenzar a tratar de describir los pasos a seguir en la solución (diagrama) nos detengamos a pensar en una estrategia de resolución. Esto quiere decir que además de detectar los datos y los objetivos debemos tratar de escribir en palabras (no en un diagrama) cómo vamos a encarar la solución.