INTRODUCCION

"Un gran descubrimiento resuelve un gran problema, pero en la solución de todo problema, hay cierto descubrimiento. El problema que se plantea puede ser modesto; pero, si pone a prueba la curiosidad que induce a poner en juego las facultades inventivas, si se resuelve por medios propios, se puede experimentar el encanto del descubrimiento y el goce del triunfo. Experiencias de este tipo, a una edad conveniente, pueden determinar una afición para el trabajo intelectual e imprimir una huella imperecedera en la mente y en el carácter"...

"How to Solve it" G. Pólya, 1945.

Desde hace varias décadas, las computadoras se han transformado en una herramienta indispensable para el comercio, la industria y la investigación científica.
Una computadora es un autómata que ejecuta un proceso de acuerdo a reglas rígidas. Este proceso normalmente es especificado por un **programa**.
Un programa consiste de una secuencia de instrucciones. Por lo general una computadora posee un repertorio muy reducido de instrucciones elementales que es capaz de entender y obedecer.
Para que una computadora pueda ejecutar un proceso especificado por un programa, éste debe contener sólo instrucciones elementales. Precisamente, el poder esencial de las computadoras es su capacidad para ejecutar enormes secuencias de acciones elementales en tiempos muy reducidos.
La actividad de generar estas secuencias de instrucciones se denomina habitualmente **programación**.
Las ideas fundamentales comprendidas en el diseño de programas pueden ser explicadas y comprendidas sin usar una computadora.
La programación es una disciplina, de las Ciencias de Computación, con muchas aplicaciones, incluyendo problemas no triviales cuya resolución constituye un desafío intelectual. En tales casos, la verdadera dificultad no reside en expresar el problema en términos de instrucciones elementales, sino en la **resolución del problema**.
En la medida que el rango de aplicación de las computadoras ha ido creciendo, abarcando áreas y problemas más grandes y complejos, el énfasis ha pasado de la **programación** al **diseño de algoritmos**. Un *algoritmo es un modelo de la resolución de un problema o más bien de una clase de problemas.*
El curso podría haberse organizado dándole énfasis a la organización de las computadoras y en cómo ejecutan programas. La **motivación primaria** será en cambio **introducir la programación como la estrategia de construir y formular algoritmos de una forma sistemática.**
Se entiende que en un curso introductorio de este tipo, los alumnos deberían ser preparados para proceder metodológicamente en el diseño de algoritmos.
Para reconocer la importancia de proceder de acuerdo a una metodología, se mostrará la complejidad inherente a la resolución de problemas, aún para ejemplos sencillos.
Las técnicas que se proponen son típicas de programación pero independientes del área de aplicación. Ningún área de aplicación será enfatizada como un fin en sí misma.
La idea es proponer una notación en la cual emerjan claramente los datos y las acciones que actúan sobre ellos.
El curso está dedicado, esencialmente, a ayudar al estudiante a desarrollar sus propias estrategias de construcción de algoritmos usando un lenguaje de diseño de algoritmos, lo suficientemente general, como para ser utilizado en los cursos de Programación como paso previo a la codificación de programas en un lenguaje de programación. Finalmente, se tratará de desarrollar la habilidad de traducir los algoritmos que resuelven problemas, de distinta complejidad, a un lenguaje de programación sencillo con sintaxis restringida.

RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS

1.1 ¿COMO ENCARAR LOS PROBLEMAS?

La verdadera dificultad no reside en expresar la solución del problema en términos de instrucciones elementales de un lenguaje de programación específico, sino en la resolución propiamente dicha del problema.
Al tratar de resolver un problema, en general, se puede repetidamente cambiar el punto de vista acerca del mismo, es decir, la forma de verlo. La comprensión de un problema probablemente sea más bien incompleta cuando se comienza a trabajar sobre él; cuando se hace algún progreso la visión del mismo se modifica y es nuevamente distinta cuando se ha obtenido la solución.
El reconocimiento que se le ha dado a la actividad de resolver problemas ha originado algunas propuestas sobre su enseñanza.
Existen diferentes modelos de resolución, Polya, Mason, Burton, Stacey, Bransford, Guzmán entre otros, que establecen cómo actuar frente a los problemas. Los modelos de Mason, Burton y Stacey, analizan los comportamientos de los sujetos reales involucrados en la resolución de problemas y sus características propias (reacciones, sensaciones, emociones, altibajos, retrocesos o inspiraciones) considerando positivos los atascos y errores que pudiesen surgir. Bransford y Guzmán proponen hábitos mentales útiles mientras que el método heurístico definido por el matemático George Polya, en 1957, describe el comportamiento de quien resuelve el problema al recorrer secuencialmente determinadas fases fundamentales pasando de una a otra sólo cuando se ha concluido con la anterior.

1.2 Método de Resolución de Problemas de Pólya

Creado por George Pólya, este plan consiste en un conjunto de cuatro pasos y preguntas que orientan la búsqueda y la exploración de las alternativas de solución que puede tener un problema. Es decir, el plan muestra cómo atacar un problema de manera eficaz y cómo ir aprendiendo con la experiencia.
La finalidad del método es que la persona examine y remodele sus propios métodos de pensamiento, de forma sistemática, eliminando obstáculos y llegando a establecer hábitos mentales eficaces; lo que Pólya denominó pensamiento productivo.
El modelo de Polya define un marco conceptual que consiste de cuatro etapas fundamentales siendo el análisis el elemento fundamental del proceso de resolución. Esta estrategia permite transformar el problema en una expresión más sencilla que se sepa resolver. Esta metodología puede pensarse como el instrumento heurístico que permite descubrir, interrelacionar y desarrollar el pensamiento crítico y reflexivo, la creatividad y capacidad de inventiva con el pensamiento computacional necesario para implementar la transformación

Comprender el problema: Se comienza por el enunciado del problema. Para ello se debe ver claramente qué es lo que se requiere, pues mal se puede responder a una cuestión que no se comprende. En esta etapa se deberá ser capaz de reconocer cuestiones tales como: principales partes del problema, la(s) incógnita(s) del mismo, es decir lo que no se conoce, los datos del problema, las condiciones y/o restricciones del mismo y se buscan patrones.
Para comprender bien el problema y precisar los límites del mismo es útil que el alumno se formule algunas productas y procure responderlas.

- ¿se entiende todo lo que dice el enunciado?
- ¿es posible replantear el problema con sus propias palabras?
- ¿reconoce los datos?
- ¿sabe a qué tiene que llegar?
- ¿tiene todos los datos necesarios?
- ¿es este problema similar a otro?

Definir un plan o Bosquejar una solución: Bosquejar una solución implica conocer qué pasos, cálculos o acciones a realizar en orden de obtener valor(es) para la(s) incógnita(s) del problema. Se puede comenzar por

considerar las partes fundamentales del problema, una vez que las mismas estén claramente reconocidas (resultado del paso anterior). Encontrar una solución no es siempre una tarea fácil, para ello influyen factores tales como conocimientos previos, buenos hábitos mentales, concentración sobre el propósito. Es importante, en esta etapa, convencerse que cada paso en el razonamiento es correcto.
Es necesario utilizar diferentes estrategias para encontrar una idea que resulte útil:

- Ensayo Error
- Buscar un patrón
- Hacer un diagrama
- Resolver una ecuación
- Buscar una fórmula
- Resolver un modelo similar más simple
- …

Ejecutar el plan: Comience por la feliz idea que lo conduce a la solución cuando esté seguro de tener el correcto punto de partida. Es necesario implementar la o las posibles estrategias seleccionadas hasta solucionar el problema completo o hasta que la misma acción provoque un cambio de rumbo. No se debe tener miedo a volver atrás. Muchas veces una nueva estrategia lleva al éxito.
En esta etapa se busca responder a preguntas tales como:

- ¿Se han empleado todos los datos?
- ¿Es posible deducir algún elemento útil de los datos?
- ¿Es posible resolver alguna parte del problema?
- ¿Se ha tenido en cuenta todas las nociones del problema?
- Los cálculos ¿son los correctos?
- …

Mirar hacia atrás, Visión retrospectiva: Esta es una excelente práctica que casi nunca se realiza. Etapa de la visión retrospectiva donde es muy importante observar qué fue lo que se hizo, verificar el resultado y el razonamiento seguido.
Cuando se ha obtenido una solución a un problema no debería cerrarse el cuaderno y pensar en cualquier otra cosa puesto que al hacerlo se pierde una fase importante e instructiva del trabajo realizado. Mirando hacia atrás, a partir de la solución completa, reconsiderando y reexaminando el resultado y el camino que condujo a él, se consolida el conocimiento y se desarrolla la habilidad para resolver problemas, puesto que no siempre un problema es examinado exhaustivamente. Con suficiente análisis y concentración casi siempre es posible mejorar la comprensión de la solución o encontrar soluciones alternativas. Pueden aparecer también casos no pensados en la primera solución. Incluso, es una práctica deseable el pensar acerca de la posibilidad de usar el resultado o el método para algún otro tipo de problema.
Seguir estos pasos no garantizará que se llegue a la respuesta correcta del problema, puesto que la resolución de problemas es un proceso complejo y rico que no se limita a seguir instrucciones paso a paso que llevarán a una solución como si fuera un algoritmo. Sin embargo, el usarlos orientará el proceso de solución del problema. Es conviene, por lo tanto, acostumbrarse a proceder de un modo ordenado, siguiendo los cuatro pasos.

1.3 PROBLEMAS

Como ser pensante el hombre se ve continuamente enfrentado a una gran diversidad de problemas. ¿Por cuál candidato votar? ¿Cuáles son las raíces de una ecuación? ¿Cómo ganar un partido de ajedrez?

Aunque estos problemas son aparentemente diferentes, todo problema puede pensarse como una discrepancia entre un estado actual o inicial y un estado deseado o final. Para pasar de un estado al otro se deben realizar determinadas tareas que son legales. A estas tareas se las denominan habitualmente **acciones.**

El proceso de hallar una solución puede pensarse como un **proceso de selección**. Cuando el problema es votar a un determinado candidato, la solución consistirá, en función de nuestra visión del mundo, de determinar cuáles son las características deseables (honestidad, credibilidad, etc.) y cuál es su programa de acción en función de las expectativas en educación, salud, vivienda, etc. Si el problema es ejecutar una operación aritmética, el resultado será un número. En cualquier caso se **está seleccionando** una respuesta adecuada.

En este sentido se puede distinguir entre dos grandes categorías de problemas: en la primera, la solución es subjetiva (el caso de elegir un candidato) mientras que en la segunda podemos ubicar aquéllos que poseen soluciones objetivas (realizar una operación aritmética).

En la primera categoría el mismo problema, dependiendo de diferentes sistemas de valores y criterios, puede dar lugar a soluciones muy dispares y aún contrapuestas según quien lo resuelva. En la segunda categoría se ubican aquellos problemas para los cuales es posible hallar soluciones que no dependan de elementos subjetivos; el planteo contiene un conjunto de condiciones o restricciones que permiten decidir si una respuesta constituye una solución o no para el problema.

La búsqueda de la solución de un problema es una tarea difícil de sistematizar. Cada problema puede presentarse en forma aparentemente aislada y frecuentemente no sabemos como encararlo.

Sin embargo, como ya se dijo, existen algunas pautas útiles que, de ser seguidas, pueden ayudarnos a enfrentar un problema. Por ejemplo:

- ✓ Establecer el problema en forma clara y entenderlo completamente.
- ✓ Clarificar cualquier ambigüedad que presente el enunciado del problema.
- ✓ Definir claramente qué se quiere hacer.
- ✓ Especificar con precisión todas las restricciones o condiciones que debe satisfacer la solución.
- ✓ Identificar claramente la información disponible.
- ✓ Explicitar toda información implícita en el planteo que pueda resultar útil.
- ✓ Encontrar una representación adecuada para esta información.
- ✓ Retomar el enunciado original ante un callejón sin salida.

Estas pautas no deben seguirse necesariamente en el orden expuesto; muchas tareas se repiten y a menudo se relee el enunciado con mayor concentración, focalizando la atención en algún aspecto que tal vez resultó inadvertido en un primer momento.

Estas pautas van a ser imprescindibles cuando intentemos escribir un algoritmo, para encontrar una solución a un problema, a partir de un enunciado o de las necesidades expresadas por un usuario.

Es frecuente que al intentar hallar la solución del problema impongamos más restricciones a la incógnita que las implicadas en el enunciado.

Por ejemplo, supongamos tener seis lápices de igual longitud y deseamos formar cuatro triángulos equiláteros de igual tamaño, sin que se crucen los lápices. Intente hallar la solución .....

Casi con seguridad, la mayor parte de los lectores han intentado formar figuras sobre una mesa o en el suelo, restringiendo el problema al plano. Probablemente esta restricción se deba a la asociación natural entre un triángulo y el plano. Intente ahora una solución en el espacio ....

El enunciado dado no establece ningún tipo de pauta para encontrar la solución; es más se aprovecha de la idea preestablecida de que la mesa es un plano, en el que se acomodan los lápices para formar los triángulos, pero el espacio es un conjunto infinito de planos.

En otras ocasiones, el enunciado impone en forma más o menos explícita y precisa, pautas para hallar la solución. En este caso, en el ejemplo dado, podría haber dicho **no se restrinja al plano**.

También resulta frecuente el caso de enunciados que presentan cierta ambigüedad. Por ejemplo, cuentan que un cazador empedernido vio 16 palomas cómodamente apoyadas en un balcón y lanzó una perdigonada, matando a 4. ¿Cuántas quedaron?. Evidentemente la respuesta inmediata es ¡12!. Sin embargo, a poco que lo pensemos,

vemos que en realidad quedaron las 4 muertas, pues las otras volaron. En fin ... la **pregunta ¿Cuántas quedaron?**, es interpretada unánimemente como **¿Cuántas quedaron vivas?**, cuya respuesta es obvia. También podría interpretarse **como ¿Cuántas quedaron en el balcón?**, en cuyo caso, la segunda respuesta es la apropiada. Esta es una situación en la cual el problema no fue bien especificado y no debe confundirse con un problema con varias soluciones.

Existen algunos otros pasos que siempre que sea posible es importante aplicarlos. Por ejemplo, eliminar toda aquella información redundante o irrelevante. Sin embargo, es muy difícil decidir desde el principio qué datos podemos descartar pues no van a ayudarnos a hallar una solución.

1.4 REPRESENTACIÓN DE PROBLEMAS – ABSTRACCIONES

Un problema se presenta en general bajo la forma de un enunciado.

No consideraremos el caso de enunciados incompletos, ambiguos o contradictorios porque en estos casos el problema no está bien especificado y deberíamos comenzar por encontrar un planteo adecuado.

La principal dificultad para hallar la solución de un problema, no es la falta de información, sino mas bien la incapacidad para utilizar convenientemente la información provista. Aún cuando un enunciado contenga información suficiente para hallar la solución del problema, es probable que haya información irrelevante, redundante o implícita.

La capacidad de la mente para manipular información es muy limitada. Una persona no es capaz de considerar muchos conceptos a la vez si no hay relación entre ellos.

Por ejemplo, observe la siguiente secuencia de dígitos durante 10 segundos:

3 0 7 1 5 9 4 6 2 8 4 5

Al dejar de mirar la secuencia, ¿Cuántos dígitos puede recordar? En promedio, las personas pueden retener y repetir unos 7 dígitos.

La mayoría de los problemas abarcan muchos más ítems de información. Un punto fundamental entonces para hallar la solución de un problema es seleccionar los elementos de información que resulten relevantes.

La información puede ir organizándose en bloques, de manera tal de ir creando una clasificación. Por ejemplo, ítems tales como manzanas, naranjas e higos, pueden agruparse en frutas. Frutas, verduras y carnes pueden conformar la clase comidas frescas. Las comidas frescas junto con los embutidos y conservas pueden agruparse en comidas, y así siguiendo.

El agrupar ítems ayuda a retener y manipular más información. Para comprobarlo, agrupe de a tres dígitos la secuencia anterior e intente nuevamente retener los números formados.

En el estudio de la lengua, se recurre frecuentemente a la frase "El mi tute mas si sede", para agrupar a todos los monosílabos que pueden acentuarse o no. Evidentemente, es muy fácil recordar esta oración, aún cuando no tiene sentido, en lugar de memorizar las 8 palabras aisladamente.

Frecuentemente resulta útil desarrollar patrones que incluyan ítems aparentemente sin relación. Los diagramas, los gráficos y las ecuaciones son ejemplos de herramientas que permiten condensar información.

El primer paso en la búsqueda de la solución es encontrar una representación adecuada para el problema, que descarte la información superflua y rescate aquélla que resulte relevante.

La técnica para encontrar un planteo más conciso del problema se conoce como **abstracción**. Una abstracción de esta clase se realiza en varias etapas:

- ✓ Transformar el enunciado original en uno tan simple como sea posible. La nueva versión no necesariamente será mas corta que la original.
- ✓ Identificar los objetos relevantes y las relaciones entre estos objetos.
- ✓ Agrupar los objetos en clases.
- ✓ Nombrar los objetos o las clases según alguna notación adecuada.
- ✓ Definir las operaciones que pueden aplicarse sobre los objetos.
- ✓ Utilizar, siempre que sea posible, una notación más abstracta que una descripción verbal.

El objetivo de la abstracción es construir una simplificación de la realidad que rescate únicamente la información relevante para hallar la solución del problema.

Intentamos construir un modelo de la realidad. Un modelo es precisamente una representación simplificada de la realidad. La construcción de un modelo es útil porque sólo son considerados los elementos relevantes y así la búsqueda de la solución no se ve entorpecida por el exceso de detalles.
En lo que sigue recapitularemos y profundizaremos algunos de los conceptos previamente definidos tales como el de ***algoritmo*** y de otros conceptos básicos relacionados.

1.4 RESOLUCIÓN DE PROBLEMAS Y COMPUTADORAS

Vamos a reconsiderar ahora las etapas de resolución de problemas, teniendo en mente, la existencia de una computadora:

1. La primer etapa, involucra la ***formulación del problema***, tal como nosotros ya la conocemos.

2. La segunda, es la ***elección de un método o procedimiento para hallar la solución*** del mismo, donde está implícito el concepto de representación ya estudiado, es decir encontrar un ***algoritmo***.

3. La tercera, llamada ***codificación***, consiste en expresar el algoritmo de manera tal que pueda ser interpretado por un ***procesador***.

4. Finalmente, la última etapa es la ***ejecución*** del procedimiento elegido para obtener la solución del problema.

1.4.1. Definiciones Asociadas

Ahora formalizaremos algunos de los conceptos importantes y relacionados:

> Los ***enunciados*** en general describen, cada uno, un determinado trabajo.

Además, definiremos de forma genérica un ***procesador*** como:

> Toda entidad capaz de comprender las acciones elementales de un algoritmo y ejecutar el trabajo indicado por el mismo.

Es importante señalar que bajo esta definición un procesador puede ser una persona, una calculadora, una computadora o una máquina cualquiera.

El procesador no puede realizar el trabajo demandado si no cuenta con los recursos necesarios. Entonces definiremos:

> El conjunto de todos los recursos necesarios para la ejecución de un trabajo, expresado a través de un algoritmo, constituye el ***ambiente*** del mismo.

Cualesquiera sea el ambiente descripto, la ejecución de un trabajo, en general, no es inmediata; ella supone un conjunto de pasos para llegar al fin perseguido. Llamaremos a cada uno de estos pasos ***acción***.

> Una ***acción*** es un evento que modifica al ambiente.

Además, cualquier procesador debe respetar la secuencia de acciones. Las acciones deben ser ejecutadas en el orden en que aparecen en el método o procedimiento seleccionado. También se hace notar que una misma acción puede aparecer, más de una vez, en la descripción de un trabajo.

El procesador tiene la capacidad de interpretar cada una de las acciones descriptas, en cuyo caso estas acciones constituyen ***acciones primitivas***.

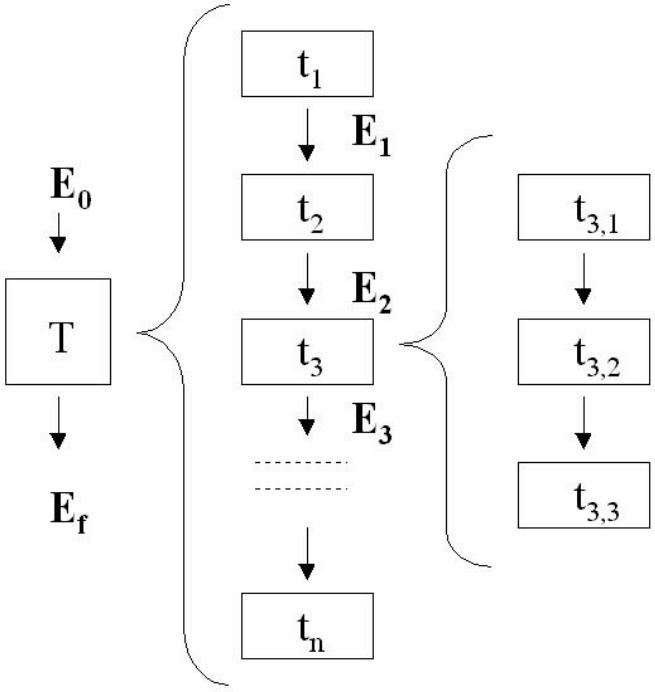
> Para un procesador dado, una ***acción es primitiva*** si su enunciado puede ser ejecutado sin información adicional.

En consecuencia, una ***acción no primitiva*** debe ser ***descompuesta en acciones primitivas***, para un procesador dado.

Existen distintas técnicas para descomponer una acción en acciones primitivas. Uno de los métodos, denominado de ***refinamiento sucesivo*** consiste en:

> Dado un trabajo T, descripto por medio de acciones no primitivas, tal que, transforma el ambiente, desde el estado inicial E_0 hasta un estado final E_f , se puede encontrar una descomposición $t_1, t_2, ..., t_n$ que constituye una secuencia de acciones primitivas que ejecuta el trabajo T.

Gráficamente:



Como se indica en la figura, cada trabajo t_i, transforma el ambiente de un estado Ei, inicial para la tarea i, en un estado E_{i+1}, final de la tarea t_i, que a su vez, es el estado inicial para la tarea t_{i+1}.

Para cada tarea t_i existen dos posibilidades, a saber:

- ✓ t_i es una acción primitiva para el procesador, dando así por finalizada la descomposición de t_i.
- ✓ t_i no es una acción primitiva para el procesador debiéndose descomponer, en una nueva secuencia $t_{i,1}, t_{i,2}, ..., t_{i,k}$.

Ahora adaptemos la definición de algoritmo a la terminología recientemente definida:

> Un ***algoritmo*** es una secuencia finita y ordenada de acciones primitivas que pueden ser ejecutadas por un procesador y que lleva a la solución de un problema.

Un algoritmo cumple con las siguientes características:

1) Finito: Un algoritmo debe terminar después de ejecutar un número finito de pasos.

2) Preciso: Cada paso en un algoritmo debe estar definido con precisión, esto es, la acción a seguir no debe ser ambigua, sino rigurosamente especificada. Considerando el ambiente en el que se trabaja, cada acción primitiva debe significar, sólo una cosa, bien determinada y sin lugar a dudas. Un algoritmo descripto en un lenguaje como inglés o español, en el cual una misma palabra puede significar varias cosas, puede no cumplir con este punto. Es por eso que se han definido los lenguajes de programación o lenguajes de computación para especificar algoritmos, ya que en ellos el significado de cada palabra es uno y sólo uno.

3) Entrada: Todo algoritmo tiene un comienzo. Se considera como entrada el conjunto de datos o información requerida para resolver un problema dado. No cualquier grupo de datos se puede considerar como entrada en el procedimiento señalado.

4) Salida: Todo algoritmo tiene un fin. La salida es un conjunto de resultados que se obtienen al aplicar el algoritmo al conjunto de datos de entrada.

5) Efectivo: Un algoritmo debe llevar a la solución del problema

> Un ***programa*** es una secuencia finita y ordenada de instrucciones escritas en algún lenguaje de programación, por medio de los cuales se logra que la computadora realice todas las operaciones o decisiones señaladas en dichas instrucciones y que llevan a la solución de un problema.

Ejemplo

Enunciado: Sumar los números 124, 59 y 3 usando una calculadora de bolsillo.

Procesador: Una persona que entienda la tarea descripta y cuente con los elementos necesarios para realizarla.

Ambiente: la calculadora de bolsillo.

Acciones Primitivas: En la notación que sigue < > significa tecla.

t_1. **Oprimir** <C> (limpiar pantalla).
t_2. **Oprimir** <dig>.
t_3. **Oprimir** <+>.
t_4. **Oprimir** <=> (muestra el resultado en el visor).

Algoritmo:

Versión 1:

t_1. Limpiar visor.
t_2. Oprimir número.
t_3. Oprimir tecla de suma.
t_4. Oprimir número.
t_5. Oprimir tecla de suma.
t_6. Oprimir número.
t_7. Oprimir tecla de igual para obtener el resultado.

Como las acciones anteriormente descriptas no son acciones primitivas deberemos efectuar el proceso de refinamiento sucesivo, en este ejemplo, que es sumamente sencillo, basta con un paso más, obteniéndose:

Versión 2 (final):

t_1. **Oprimir** <C> (limpiar pantalla).
$t_{2,1}$. **Oprimir** <1>.
$t_{2,2}$. **Oprimir** <2>.
$t_{2,3}$. **Oprimir** <4>.
t_3. **Oprimir** <+>.
$t_{4,1}$. **Oprimir** <5>.
$t_{4,2}$. **Oprimir** <9>.
t_5. **Oprimir** <+>.
t_6. **Oprimir** <3>.
t_7. **Oprimir** <=> (muestra el resultado en el visor).

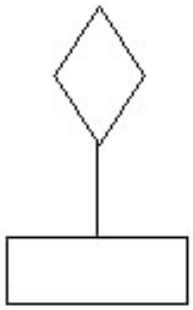
El procesador debe ser también capaz de interpretar órdenes que **no modifican el ambiente**, tal como la ***repetición***

La **repetición** permite al procesador ejecutar una secuencia de acciones más de una vez.

Por ahora la forma de este tipo de acción será:

REPETIR <número> [<acciones_primitivas>]

1.4.1. Un ejemplo completo



Enunciado: Se desea dibujar un farol con pedestal del siguiente tipo:
Procesador***:*** Una tortuga con habilidad para moverse y dibujar sobre el plano.
Ambiente***:*** Una hoja de papel.

Acciones Primitivas:

DERECHA <grados>
IZQUIERDA <grados>
ADELANTE <número_de_pasos>.

La acción de repetición que puede interpretar el procesador es la dada anteriormente:

REPETIR <número> [<acciones_primitivas>]

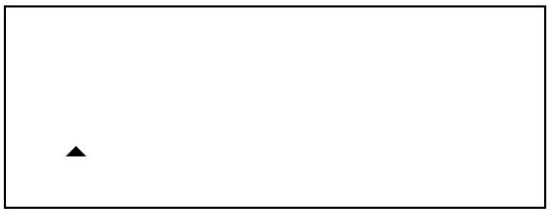
Algoritmo:
Situación inicial del ambiente:

Versión 1:

T_1: Dibujar la base, que es un rectángulo.
T_2: Dibujar el poste, que es una línea.
T_3: Dibujar el farol, que es un rombo.

Pero ninguna de estas descripciones es una acción primitiva que pueda ser entendida por el procesador, entonces concentremos nuestra atención en T_1.

Versión 2:
Teniendo en cuenta la situación inicial graficada,

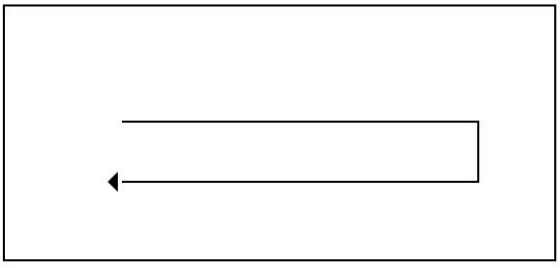


vemos que una posible forma de dibujar la base es suponer que donde está la tortuga es la esquina superior izquierda, entonces deberíamos:

$T_{1,1}$: Dibujar el borde superior del rectángulo.

$T_{1,2}$: Dibujar el borde derecho del rectángulo.

$T_{1,3}$: Dibujar el borde inferior del rectángulo.



$T_{1,4}$: Dibujar el borde izquierdo del rectángulo.

Pero las acciones enumeradas en las tareas $T_{1,1}$, $T_{1,2}$, $T_{1,3}$ y $T_{1,4}$ aún no son acciones primitivas entendibles por el procesador, entonces continuando con nuestra técnica de desagregación, por refinamientos sucesivos, obtenemos una nueva versión.

Versión 3:
Donde **$T_{1,1}$ dibujar el borde superior del rectángulo** implicará:

Como la posición inicial de la tortuga es ▲, para empezar a dibujar la línea superior del rectángulo, primero deberemos girarla hacía la derecha 90° para que quede ▶ y, luego, ir hacía adelante un cierto número de pasos, por ejemplo 40, lo cual expresado en acciones primitivas es:

$T_{1,1,1}$ DERECHA 90
$T_{1,1,2}$ ADELANTE 40.

Ahora, debemos continuar con la acción **$T_{1,2}$ dibujar el borde derecho del rectángulo,** para ello lo primero a recordar es la posición en que quedó la tortuga, tal es ▶ y lo que nosotros queremos ahora es que quede ▼ para que luego pueda avanzar dibujando el lado derecho del rectángulo en por ejemplo 10 pasos; luego con un razonamiento análogo al anterior obtenemos:

$T_{1,2,1}$ DERECHA 90
$T_{1,2,2}$ ADELANTE 10

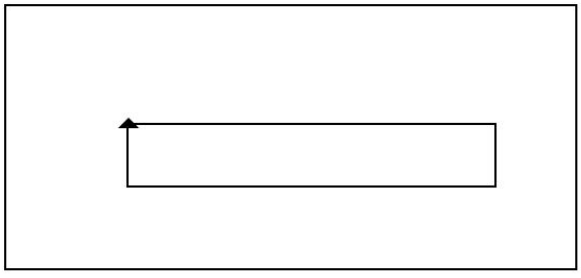
Debemos realizar un razonamiento similar al realizado para desagregar la tarea $T_{1,1}$ pero ahora debemos tener en cuenta que al finalizar el dibujo del borde derecho la tortuga quedó ▼ y nosotros la necesitamos ◀ para luego dibujar el borde inferior del rectángulo, entonces tendremos:

$T_{1,3,1}$ DERECHA 90
$T_{1,3,2}$ ADELANTE 40

Por último, la acción **$T_{1,4}$ dibujar el borde izquierdo del rectángulo**, quedará expresada en acciones primitivas como:

$T_{1,4,1}$ DERECHA 90
$T_{1,4,2}$ ADELANTE 10

Con lo que obtuvimos el dibujo deseado, el pedestal de la farola, con la tortuga nuevamente en la posición inicial:



Agrupando todas las acciones primitivas que componen la acción T_1, obtenemos el algoritmo completo para realizar dicha actividad, que queda como sigue:

$T_{1,1,1}$ DERECHA 90
$T_{1,1,2}$ ADELANTE 40.
$T_{1,2,1}$ DERECHA 90
$T_{1,2,2}$ ADELANTE 10
$T_{1,3,1}$ DERECHA 90
$T_{1,3,2}$ ADELANTE 40
$T_{1,4,1}$ DERECHA 90
$T_{1,4,2}$ ADELANTE 10

Observando el algoritmo vemos que existe una secuencia de cuatro acciones primitivas que se repite dos veces, como es de esperar, dado que se deben dibujar los dos bordes, inferior y superior y los otros dos, izquierdo y derecho del rectángulo, entonces es posible simplificar el algoritmo escribiendo una sola vez dicha secuencia y usando la acción primitiva de repetición, con lo cual el algoritmo final para la actividad T_1 es:

$T_{1,1}$ **REPETIR** 2 [DERECHA 90 ADELANTE 40 DERECHA 90 ADELANTE 10]

más las dos actividades que aún restan descomponer:

T_2 Dibujar el poste, que es una línea
T_3 Dibujar el farol, que es un rombo.

Versión 4:

$T_{1,1}$ **REPETIR** 2 [DERECHA 90 ADELANTE 40 DERECHA 90 ADELANTE 10]
T_2 Dibujar el poste, que es una línea.

La idea ahora es trabajar sobre la acción T_2 descomponiéndola en acciones primitivas, para ello debemos recordar que al concluir de dibujar la base del farol la tortuga estaba en su posición inicial, ▲, lo que deseamos es llevarla hasta la mitad del borde superior de la base y desde esa posición comenzar a dibujar el poste, esto implicará:

$T_{2,1}$ girar la tortuga hacía la derecha 90°
$T_{2,2}$ avanzar 20 pasos (la base tenía un ancho de 40)
$T_{2,3}$ girar la tortuga a la izquierda 90 °
$T_{2,4}$ avanzar, por ejemplo, 60 pasos.

Lo cual expresado en acciones primitivas queda:
$T_{2,1}$ DERECHA 90
$T_{2,2}$ ADELANTE 20
$T_{2,3}$ IZQUIERDA 90
$T_{2,4}$ ADELANTE 60

Luego la versión 4 del algoritmo quedó:
$T_{1,1}$ **REPETIR** 2 [DERECHA 90 ADELANTE 40 DERECHA 90 ADELANTE 10]
$T_{2,1}$ DERECHA 90
$T_{2,2}$ ADELANTE 20
$T_{2,3}$ IZQUIERDA 90
$T_{2,4}$ ADELANTE 60
T_3 Dibujar el farol, que es un rombo.

donde la única acción que no es primitiva es T_3.

Versión 5:

Centrándonos sólo en la actividad T_3 vemos que el farol es un rombo, para poder dibujarlo debemos, primero, ubicar la tortuga que quedó en la posición ▲, girándola 45° y , luego, dibujar un cuadrado. El dibujo del cuadrado implica las acciones repetidas de dibujar un lado, por ejemplo, 20 pasos y girar a la derecha un ángulo de 90° ¿Cuántas veces se realizarán estas dos últimas acciones? Cuatro veces, entonces ya nos damos cuenta que podemos usar una acción primitiva de repetición, luego la tarea T_3 desagregada quedará:

$T_{3,1}$ IZQUIERDA 45°
$T_{3,2}$ REPETIR 4 [ADELANTE 20 DERECHA 90]

En consecuencia**, la versión final del algoritmo que resuelve el problema planteado** es:

$T_{1,1}$ **REPETIR** 2 [DERECHA 90 ADELANTE 40 DERECHA 90 ADELANTE 10]
$T_{2,1}$ DERECHA 90
$T_{2,2}$ ADELANTE 20
$T_{2,3}$ IZQUIERDA 90
$T_{2,4}$ ADELANTE 60
$T_{3,1}$ IZQUIERDA 45°
$T_{3,2}$ **REPETIR** 4 [ADELANTE 20 DERECHA 90]