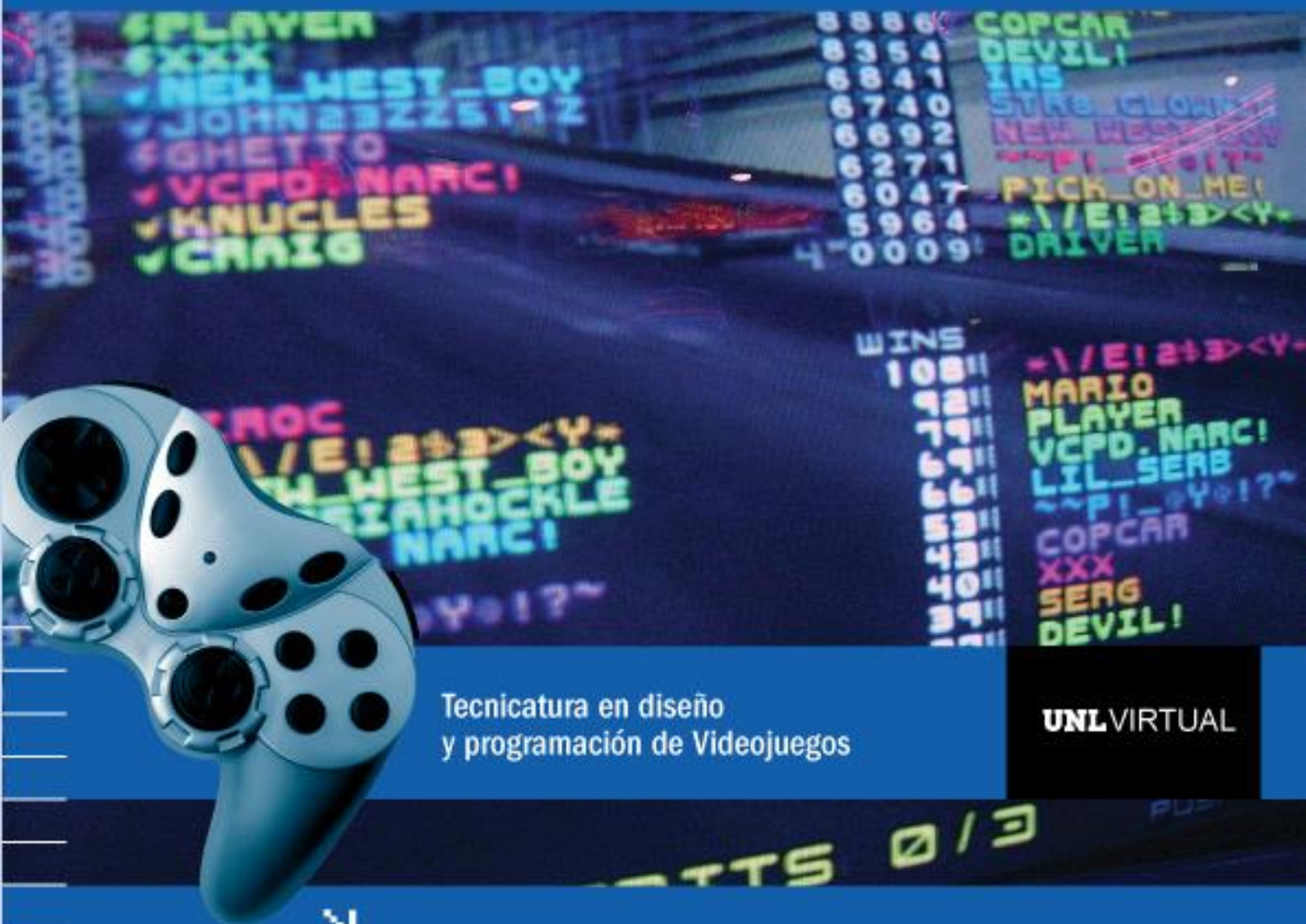




UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS



Tecnicatura en diseño
y programación de Videojuegos

UNL VIRTUAL



Introducción a la programación

Unidad Temática 1

Preparando las herramientas

- **Objetivo:** Interiorizarse en los conceptos básicos relativos a la creación de programas y comprender un típico entorno de desarrollo en C++.
- **Temas:** Introducción, Conceptos básicos: Algoritmo, Programa, Programación, Resolución de Problemas, Lenguaje de programación, Compilador, Depuración, Instalación y uso del compilador (IDE DevC++), Creando un espacio de trabajo, Estableciendo un estándar en el desarrollo.

INTRODUCCIÓN

Tal como lo indica el nombre de la materia y el apunte, en este espacio dedicaremos nuestro tiempo a aprender las nociones básicas de la programación, informáticamente hablando. Lo cual nos lleva a la pregunta elemental. ¿Qué es programar? Programar es decirle a la computadora qué es lo que queremos que haga, lo cual no es tan simple, porque además hay que decirle cómo tiene que hacerlo y, para agregar dificultad, se lo tenemos que decir en los términos que ella entienda. Esto involucra, además de escribir en un lenguaje estricto, expresar nuestras ideas con un pensamiento particular. Este pensamiento se denomina Pensamiento Lógico, y es en lo primero que debemos enfocarnos para aprender programación.

El pensamiento lógico es una forma de entender y razonar el mundo que nos rodea de una manera fáctica y coherente, esto quiere decir que no se aceptan contradicciones ni puntos grises. Como ya deben saber, las computadoras procesan a partir de un lenguaje binario (1 y 0), por lo que sólo comprenden una orden si es cierta o falsa y a eso hace referencia el pensamiento lógico. Lo importante de este proceso es entender que el pensamiento lógico **no se aprende**, sino que se **desarrolla**, cada uno por su cultura, formación o simple curiosidad puede tener mayor o menormente desarrollado su pensamiento lógico, es por ello que no hay que preocuparse si en un principio cuesta entender de qué estamos hablando, todo es cuestión de perseverancia y abrirse a nuevas formas de enfocar la realidad.

RESOLUCIÓN DE OBJETIVOS CON PENSAMIENTO LÓGICO

Quizás no lo mencionamos antes, pero la programación nos permite realizar tareas (resolver problemas) utilizando la computadora como herramienta. Cabe hacer esta aclaración, porque el pensamiento lógico es la forma en la que deberemos construir nuestras **estrategias** para resolver esos problemas. Del pensamiento lógico nosotros queremos desarrollar dos habilidades que nos permitirán resolver cualquier tipo de problemas, y estas son: la **abstracción** y la **operacionalidad**.

La abstracción incluye técnicas como simplificación, la generalización y la identificación. Ya veremos esto en breve. Las habilidades operacionales suponen la definición de soluciones en términos de un conjunto de pasos que deben ejecutarse en un orden determinado para alcanzar un objetivo. Y eso es lo que veremos a continuación.

La forma en la que la computadora puede entender lo que queremos que haga por nosotros es enviarle instrucciones de cómo realizar algo, la forma de enviarle instrucciones no es más que una lista de acciones, tal cual si fuera una receta de cocina.

Vamos a suponer que queremos mostrarle a alguien cómo preparar un té con un saquito. De forma concisa podemos decir:

Calentar el agua, poner un saquito de té en una taza, echar el agua en la taza, esperar de 3 a 4 minutos, retirar el saquito. Listo.

Esta serie de instrucciones son, seguramente, perfectamente claras para cualquier persona... ¿Lo son? Imaginemos que la persona que quiere preparar el té está un poco más dubitativa y nuestras instrucciones no alcanzan. ¿Cómo las podemos mejorar?

Abrir la canilla, poner agua en la pava o algún recipiente para calentar, la cantidad debe ser igual o aproximada a la cantidad requerida de té para preparar, colocar en la hornalla o sistema similar hasta los 75 grados Celsius, verter el agua dentro de una taza que posea dimensiones necesarias para abarcar el agua caliente, poner el saquito de té dentro de la taza inmerso en el agua, tapar por 3 minutos, pasados los 3 minutos quitar la tapa, retirar el saquito y beber.

Seguramente esta serie de pasos son muchos más claros para cualquiera y hasta podríamos decir que son casi “de computadora”. Podemos ser infinitamente precisos a la hora de detallar las instrucciones.

Esto que acabamos de hacer (de detallar las instrucciones) se denomina **Algoritmo**. Un algoritmo es un conjunto de operaciones sistemáticas y **ordenadas** que permiten

hallar la solución de un tipo de problema y es a lo que hacíamos mención con operacionalidad.

Remarquemos como importante la parte que dice “de un tipo de problema”. Es decir, no podemos resolver todos los problemas de la vida con algoritmos, pero en la informática, todos los problemas se resuelven empleando algoritmos y para aprender a expresar un problema cómo un algoritmo debemos utilizar el pensamiento lógico. Es aquí donde retomamos nuevamente el concepto de lógica. Volviendo a la instrucción de la taza de té, no podemos decirle a una computadora que prepare un té, pero hay otro tipo de problemas que sí puede resolver.

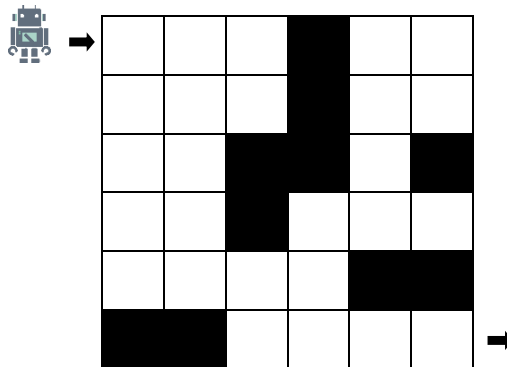
Ya vimos que necesitamos explicar la solución de un problema mediante una secuencia de pasos. Pero ¿Cómo hacemos la secuencia? Aquí entra en juego la abstracción, que es el otro conjunto de habilidades del pensamiento lógico. Cómo ya mencionamos, podemos describir en la abstracción 3 conjuntos de habilidades que nos servirán para resolver cualquier tipo de problemas, desde el más simple hasta el más complejo. Y estas habilidades son: **Simplificación**, quiere decir, interpretar un problema de una manera precisa, excluyendo problemas ajenos y que no nos atañen. **Generalización**, quiere decir, la habilidad de crear una solución común a todos los problemas de ese tipo. **Identificación**, quiere decir, la habilidad de nombrar o identificar las partes de cada problema.

MANEJANDO A UN ROBOT

Vamos a suponer que tenemos que programar un robot para que éste pueda pasar a través de un laberinto. Las instrucciones se las debemos dar todas juntas y esperar que las ejecute hasta finalizar, no puede corregir sobre una vez que empieza ni finalizar antes de tiempo.

El conjunto de instrucciones que podemos darle al robot es limitada y bien precisa

Este es el escenario:



El robot ingresa por la flecha de la esquina superior izquierda y sale por la flecha de la esquina inferior derecha. Los bloques negros son paredes y debe esquivarlos.

Las instrucciones que podemos darle son:

- Avanzar
- Retroceder
- Girar hacia la derecha (90 grados)
- Girar hacia la izquierda (90 grados)

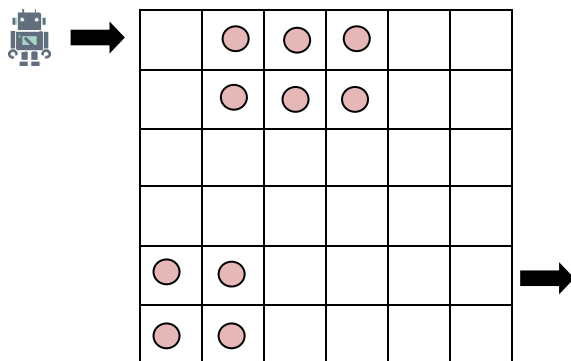
Una solución posible es avanzar, avanzar, girar derecha, avanzar, avanzar, avanzar, avanzar, girar izquierda, avanzar, avanzar, girar derecha, avanzar, girar izquierda, avanzar, avanzar, avanzar.

Como podemos ver, hay muchas soluciones posibles que nos sirven para cumplir nuestro objetivo. Cada quien, luego, podría haber pensado, en base a su intuición, una forma distinta de hacer llevar el robot hacia la salida. Retomando con los conceptos que mencionamos, cada secuencia de pasos construimos para darle a nuestro robot la llamamos **algoritmo**, aunque cumpla correctamente o no el objetivo.

¿Qué diferencia notamos entre las instrucciones que teníamos a la hora de preparar el té y las que tenemos para poder programar el robot?

A simple vista notamos que las instrucciones para preparar el té son significativamente más complejas y muy generales, por ejemplo, no podríamos haber dicho al robot, “avanzar hasta la salida” o “esquivar pared”. Ambas instrucciones son muy complejas para ser interpretadas. Cada instrucción que le dimos a nuestro robot recibe el nombre de **primitiva**.

Vamos a pensar otro escenario para nuestro robot.



Esta vez el objetivo es no solo salir del laberinto, sino juntar todas las bolitas rojas. Las primitivas que tenemos son:

- Avanzar
- Retroceder
- Girar hacia la derecha (90 grados)
- Girar hacia la izquierda (90 grados)
- Juntar bolita roja

Pese a que no hay paredes que nos obliguen a encaminarnos por algún sector del laberinto, es muy probable que la mayoría pensemos un algoritmo para cumplir el objetivo que involucre juntar las bolitas de arriba (el grupo de 6) y luego todas las de abajo (el grupo de 4), o viceversa, pero manteniendo la idea resolver primero una parte del laberinto y luego otra. Esta idea de organizar un procedimiento para llegar al objetivo se llama **estrategia**. La estrategia consiste en encontrar un método que nos permita llegar a resolver el problema planteado. No importa cuán obvia parezca, todos los algoritmos se construyen a partir de una estrategia. Incluso el primer laberinto se resolvió a partir de una estrategia. ¿Cuál les parece que fue?

Las estrategias son personales y tienen que ver con la intuición y la forma organizar los recursos. A medida que vamos sumando práctica, las estrategias que pensemos son cada vez mejores y más eficientes.

Ahora que sabemos lo que es una estrategia, podemos analizar un poco más el algoritmo. Vamos a suponer la siguiente solución:

Avanzar, avanzar, juntar bolita, avanzar, juntar bolita, avanzar, juntar bolita, girar derecha, avanzar, juntar bolita, avanzar, juntar bolita, avanzar, juntar bolita. Girar izquierda, avanzar, avanzar, avanzar. Juntar bolita, avanzar, juntar bolita, girar derecha, avanzar, juntar bolita, girar derecha, avanzar, juntar bolita. Girar derecha, avanzar, avanzar, avanzar, avanzar, avanzar, avanzar.

Nuestra estrategia fue:

Juntar las bolitas de arriba, luego, juntar las bolitas de abajo.

Si pudiéramos dividir nuestro algoritmo en diferentes etapas, podríamos distinguir 4 etapas:

- Juntar las bolitas de arriba:

Avanzar, avanzar, juntar bolita, avanzar, juntar bolita, avanzar, juntar bolita, girar derecha, avanzar, girar derecha, juntar bolita, avanzar, juntar bolita, avanzar, juntar bolita.

- Dirigirnos hacia las bolitas de abajo:

Girar izquierda, avanzar, avanzar, avanzar.

- Juntar bolitas de abajo:

Juntar bolita, avanzar, juntar bolita, girar derecha, avanzar, juntar bolita, girar derecha, avanzar, juntar bolita

- Dirigirnos a la salida:

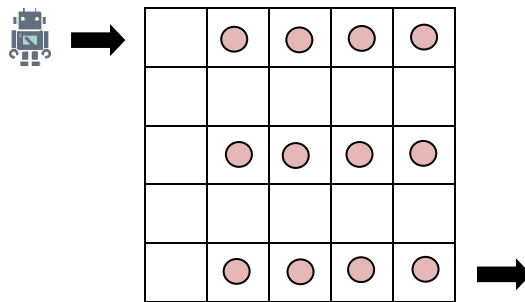
Girar derecha, avanzar, avanzar, avanzar, avanzar, avanzar, avanzar, avanzar.

Esta acción de poder localizar diferentes etapas en nuestra estrategia tiene que ver con la **identificación**, y como mencionamos antes, es una de las habilidades de la abstracción.

A partir de este nuevo concepto, vamos a proponer un conjunto de pasos comunes para resolver cualquier algoritmo:

- Identificar estrategia de resolución
- Identificar las partes que componen la estrategia
- Implementar la estrategia general y las partes que la componen

Vamos a analizar un tercer laberinto. El objetivo es el mismo, salir del laberinto habiendo juntado todas las bolitas.



Las primitivas son:

- Avanzar
- Retroceder
- Girar hacia la derecha (90 grados)
- Girar hacia la izquierda (90 grados)
- Juntar bolita roja

Vamos a emplear los pasos que mencionamos para realizar el algoritmo.

Identificar estrategia de resolución:

Juntar las bolitas de arriba fila por fila y luego salir del laberinto. (Cada uno podría interpretar su propia estrategia).

Identificar las partes que componen la estrategia:

Juntar las bolitas de arriba, bajar hasta la fila del medio, juntar las bolitas del medio, bajar hasta la fila de abajo, juntar las bolitas de abajo.

Implementar la estrategia general y las partes que la componen:

Juntar pelotitas de arriba: Avanzar, avanzar, juntar pelotita, avanzar, juntar pelotita, avanzar, juntar pelotita, avanzar, juntar pelotita, retroceder, retroceder, retroceder.

Dirigirse hacia la fila del medio: Girar derecha, avanzar, avanzar, girar izquierda.

Juntar pelotita del medio: juntar pelotita, avanzar, juntar pelotita, avanzar, juntar pelotita, avanzar, juntar pelotita, retroceder, retroceder, retroceder.

Dirigirse hacia la fila de abajo: Girar derecha, avanzar, avanzar, girar izquierda.

Juntar pelotita de abajo: juntar pelotita, avanzar, juntar pelotita, avanzar, juntar pelotita, avanzar, juntar pelotita, avanzar.

A partir de la estrategia propuesta podemos ver bien distinguidas las partes que integran la estrategia. Pero también es posible ver cierto patrón (un patrón es la recurrencia de objetos o procesos), sobre todo visualmente en laberinto. El hecho de que las filas de bolitas tengan la misma cantidad y se ubiquen a la misma distancia una de las otras es un llamado de atención para nuestra percepción. A la hora de implementar la solución del algoritmo esto se ve como una repetición de primitivas, la acción de “juntar pelotitas” en cada una de las filas es casi idéntica, si no fuera porque la fila superior incluye la parte de ingresar al laberinto y la parte inferior la de salir del laberinto, pero el conjunto de primitivas que permiten juntar las pelotitas es idéntico en cada parte:

Juntar pelotita, avanzar, juntar pelotita, avanzar, juntar pelotita, avanzar, juntar pelotita.

Lo mismo pasa con las dos etapas de la estrategia que utilizamos para posicionarnos en la siguiente fila, en este caso incluso, el conjunto de primitivas es idéntico. Si le pusiéramos nombre a este grupo de primitivas, podríamos hacer más claro nuestra estrategia.

Vamos a nombrar al conjunto de primitivas que juntan las pelotitas y avanzan como “avanzar juntando pelotitas” y al conjunto de primitivas que nos posicionan en la siguiente fila como “posicionarse en siguiente fila”.

Ahora pensemos que este conjunto de primitivas se convierten ahora en una nueva primitiva, aunque en realidad recibe el nombre de **procedimiento**. Ahora podemos agrandar la lista de primitivas y procedimientos que podemos usar:

- Avanzar
- Retroceder
- Girar hacia la derecha (90 grados)
- Girar hacia la izquierda (90 grados)
- Juntar bolita roja
- Avanzar juntando pelotitas (procedimiento)
- posicionarse en siguiente fila (procedimiento)

Entonces nuestra estrategia queda así:

Juntar pelotitas de arriba: Avanzar, avanzar, *Avanzar juntando pelotitas*, retroceder, retroceder, retroceder.

Dirigirse hacia la fila del medio: *Posicionarse en siguiente fila*.

Juntar pelotita del medio: *Avanzar juntando pelotitas*, retroceder, retroceder.

Dirigirse hacia la fila de abajo: *Posicionarse en siguiente fila*.

Juntar pelotita de abajo: *Avanzar juntando pelotitas*, avanzar.

De esta manera nuestra estrategia queda mucho más legible. Buscar patrones en la estrategia es parte del pensamiento lógico y nos permite crear estrategias comunes para problemas comunes, es lo que anteriormente mencionamos como **Generalización**. Finalmente la **Simplificación** la aplicamos cuando utilizamos procedimientos sin incluir situaciones particulares. Es decir, por ejemplo, *Avanzar juntando pelotitas* no contempla que la fila superior requiere dos movimientos extras y que difieren de las filas restantes, es preferible aplicar una solución generaliza y tratar los casos particulares de manera específica.