

Pensamiento Computacional

Descomposición de
problemas

Clase 6

Ingeniería en ciberseguridad

La excelencia no se improvisa



CLASE 6

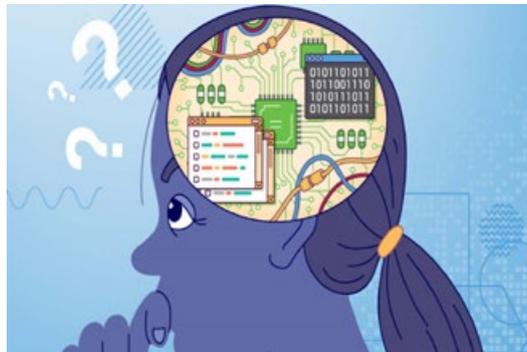
2.2 Descomposición de problemas

¿Qué es el Pensamiento Computacional?

Consiste en resolver problemas cotidianos utilizando los conceptos fundamentales de la programación, cuyas soluciones pueden representarse mediante pasos o instrucciones.

Figura 1

Pensamiento computacional



Fuente: Del Angel (2023)

Etapas del pensamiento computacional

A continuación, las etapas:

- **Descomposición:** permite dividir un problema en subproblemas más pequeños y manejables, cuyas soluciones combinadas contribuirán a la solución final del problema original.
- **Abstracción:** permite centrarse en las características más importantes del problema, captando su esencia. Esto se logra al invisibilizar las características no fundamentales y destacar, a su vez, los rasgos más relevantes, desarrollando así una representación simplificada del problema.
- **Reconocimiento de patrones:** permite buscar similitudes entre los distintos subproblemas dentro de una misma situación, facilitando la búsqueda de soluciones.
- **Diseño de algoritmos:** permite la creación de un programa ejecutable que resuelva el problema. Este programa debe ser evaluado en cuanto a su efectividad y ausencia de errores.

Figura 2

Fases pensamiento computacional



Fuente: Batistelli (2022)

Descomposición

Es el proceso de dividir un problema en partes más pequeñas o sencillas. Descomponer un problema facilita su resolución al reducir su complejidad. En esencia, toda descomposición implica:

1. Identificar las partes de algo.
2. Dividir algo en partes más pequeñas.

Ejemplo:

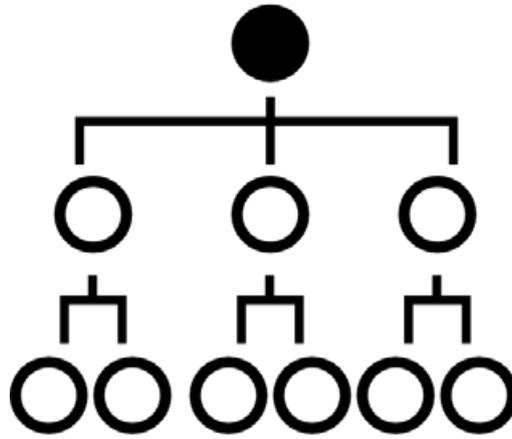
Preparar la fiesta de graduación de los estudiantes de la PUCE.

Dividimos el evento en tareas más pequeñas:

- Armado del programa de actividades.
- Difusión del evento.
- Preparación.
- Ensayo de actividades artísticas.
- Preparación del servicio de aperitivos y bebidas.
- Preparación del sitio logístico.

Figura 3

Descomposición



Fuente: Uscudum (s. f.)

Abstracción

Un proceso de **abstracción permite definir la esencia de algo**. Se basa en identificar lo que es importante sin preocuparse por los detalles. Como resultado de la abstracción, podemos determinar los aspectos más relevantes de un problema y elaborar una representación simplificada del mismo. Se utiliza para mostrar, mediante ejemplos simples, el estudio y la comprensión de sistemas complejos.

Ejemplos:

- En música, los pentagramas son soportes de escritura de piezas musicales y pueden entenderse como abstracciones que representan sonidos en el tiempo.
- En una agenda, se utiliza la abstracción para representar una semana en términos de días y horas, lo que la convierte en un objeto útil para organizar el tiempo personal.

Figura 4

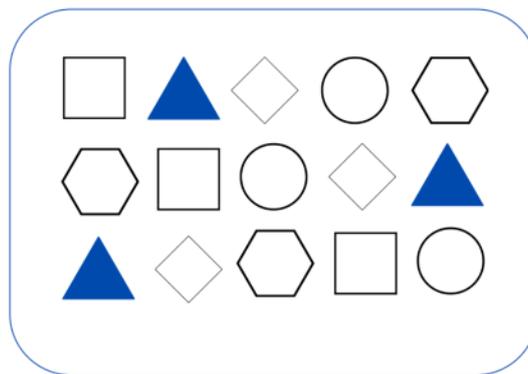


2.3 Reconocimiento de patrones

En el reconocimiento de patrones, el objetivo es **encontrar similitudes entre los distintos subproblemas dentro de una misma situación**. De esta manera, se utiliza el reconocimiento de problemas similares que han sido resueltos con anterioridad y se aplica directamente (generalización). Cuantos más patrones se reconozcan, más fácil y rápido será el desarrollo de un proyecto. La generalización es una tarea relacionada con la identificación de patrones, semejanzas y conexiones, que permite explotar las características encontradas.

Figura 5

Patrones



Fuente: Autor

Funcionamiento del reconocimiento de patrones

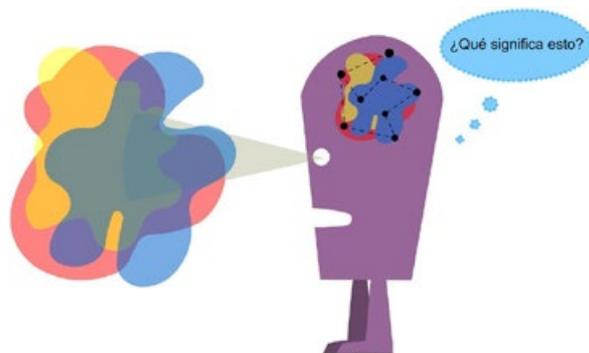
Con el reconocimiento de patrones se busca crear reglas y resolver problemas más generales. A continuación, te indicamos los pasos a seguir:

1. Recopilar datos.
2. Organizar los datos.
3. Analizar los datos para detectar patrones y tendencias.
4. Usar los patrones identificados.

El reconocimiento de patrones no solo se utiliza en áreas de la computación; también puede aplicarse en matemáticas, medicina y otras disciplinas.

Figura 6

Reconocimiento de patrones



Nota. Licencia Creative Commons

Ejemplo de reconocimiento de patrones

Una persona está a cargo de un campo con animales, y hay comida suficiente y especial para cada uno. Las instrucciones para que un operario alimente a los animales son las siguientes:

- Para alimentar al perro, poner la comida del perro en el plato del perro.
- Para alimentar al pollo, poner la comida del pollo en el plato del pollo.
- Para alimentar al conejo, poner la comida del conejo en el plato del conejo.

¿Qué patrón se puede visualizar? (un proceso de generalización): Para **alimentar** al <animal>, **poner la comida** del <animal> en el **plato** del <animal> (Bordignon & Adrián, 2020).

Diseño de algoritmos

Su objetivo es crear un conjunto de instrucciones lógicas y precisas que nos permitan resolver el problema de manera eficiente y efectiva. Los pasos de esta etapa son los siguientes:

1. Definir el problema específico y los resultados deseados.
2. Identificar los datos necesarios para resolver el problema.
3. Determinar el orden de los pasos necesarios para llegar al resultado deseado.
4. Diseñar el algoritmo paso a paso.

A continuación, veremos algunos ejemplos de algoritmos:

Figura 7

Diseño de algoritmo – basado en texto

Problema:

Preparar una sopa instantánea en el microondas.

Planteamiento de la solución:

1. Iniciar algoritmo.
2. Destapar el envase de sopa.
3. Agregar una taza de agua al envase de sopa.
4. Introducir el envase en el horno microondas.
5. Programar el horno microondas por 3 minutos.
6. Fin del algoritmo.

Figura 8

Sopa instantánea



Nota. Licencia Creative Commons

Diseño de algoritmo – usando un diagrama de flujo

Problema:

Encender una lámpara de velador.

La imagen muestra un diagrama con el algoritmo de solución para encender una lámpara.

1. Identificamos cuál es el problema: la lámpara no funciona.
2. Realizamos una comprobación: ¿Está enchufada?
 - o Si lo está, continuamos a la siguiente comprobación.
 - o Si no lo está, hay que enchufarla.
3. Otra comprobación: ¿El foco está quemado?
 - o Si está quemado, reemplazar el foco por uno nuevo.
 - o Si no está quemado, hay que comprar una nueva lámpara.

Estos diagramas se conocen como “diagramas de flujo” y se explicarán más adelante en el apartado de Pseudocódigo y flujogramas.

Figura 9



Nota. Licencia Creative Commons

Ejemplo práctico: Hacer caminar a un robot

Una vez que hemos aprendido los conceptos principales sobre las fases del pensamiento computacio-

nal, vamos a ver un ejemplo práctico.

Problema: Se requiere que un robot camine siguiendo un camino.

Descomponer

¡Manos a la obra! Usando las cuatro fases del pensamiento computacional, comenzamos por descomponer.

La construcción del robot se dividirá en lo siguiente:

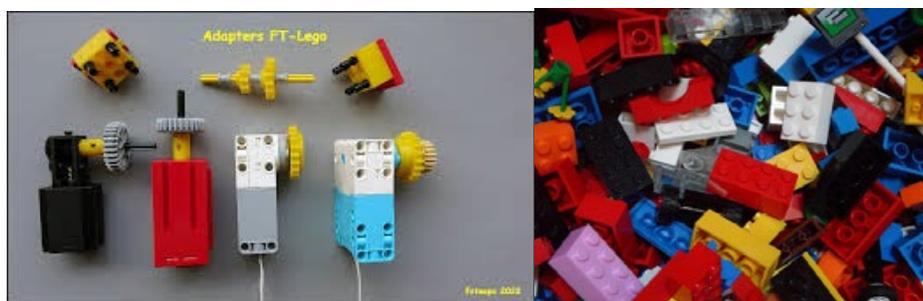
- Identificar los materiales necesarios para construir el robot.
- Diseñar y planificar el modelo del robot.
- Armar el cuerpo del robot y sus piezas individuales, como ruedas, motores, sensores, etc.
- Probar el robot para asegurarse de que todo funcione correctamente.

La programación del robot se dividirá así:

- Instalación o verificación de sensores en la parte inferior para que el robot sea capaz de seguir la línea negra.
- Programación del robot mediante la escritura de código que le permita leer la información de los sensores y determinar la dirección en la que debe moverse para seguir la línea negra.

Figura 9

Legos Motor y piezas



Nota. Licencia Creative Commons

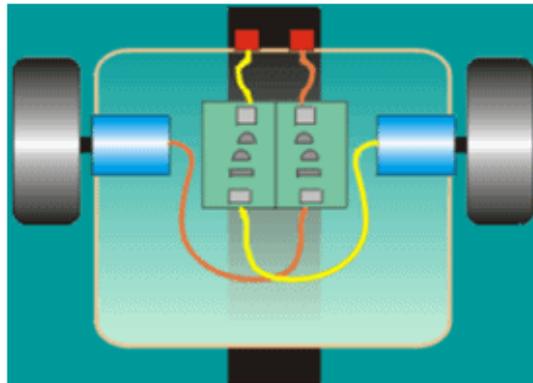
Abstracción

Una vez que hemos dividido el problema en tareas más pequeñas, pasamos a la abstracción. En esta

etapa, nos enfocamos en lo más importante: identificar qué tipo de sensores necesita el robot para detectar la línea negra y cuál será la programación para mover el robot según los datos que envíen los sensores. Los detalles del robot, como el tamaño, la forma o la velocidad a la que se moverá, se ignoran por el momento.

Figura 10

Sensores de un robot



Fuente: TecnoSalva (2020)

Reconocimiento de patrones

Continuamos con esta fase del reconocimiento de patrones: ¡los detalles cuentan! Supongamos que el robot está construido y programado, pero no sigue la línea negra. Para resolver este problema, necesitamos recopilar datos como la forma y el color de la línea negra, la dirección en la que se dibuja la línea, la luz ambiental y otros factores que puedan influir en el reconocimiento por parte de los sensores.

Se organizan los datos y se procede a buscar patrones entre el comportamiento del robot y los datos recogidos. Por ejemplo, podemos entender que el robot no detecta la línea negra cuando hay poca iluminación. Una vez que se ha identificado el patrón, podemos ajustar la luz para lograr que el robot se mueva.

Figura 11

Reconocimiento de patrones

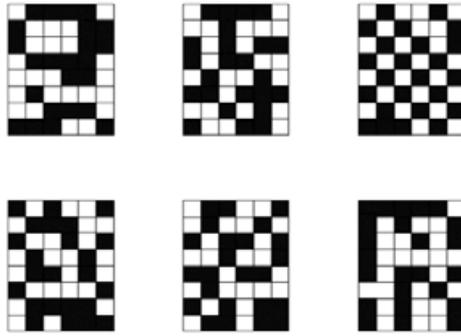


Figura 2. Patrones Aleatorios.

Fuente: Jiménez (s. f.)

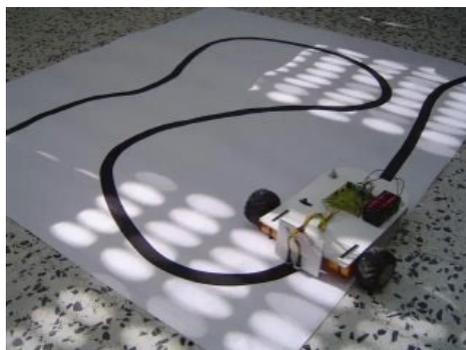
Diseño del algoritmo

Finalmente, llegamos a la cuarta fase, donde se crea el algoritmo y se programa. El algoritmo debería contemplar:

1. Iniciar los sensores y asegurarse de que estén funcionando correctamente.
2. Detectar la línea negra utilizando los sensores.
3. Ajustar la dirección del robot en función de la posición detectada de la línea negra.
4. Mover el robot en la dirección adecuada y ajustar la velocidad según sea necesario.
5. Repetir el proceso hasta que el robot llegue al final de la línea negra.

Figura 12

Robot seguidor de líneas



Fuente: TecnoSalva (2020)

Actitudes del pensador computacional

Las siguientes son las actitudes que hacen a una persona resolutiva en el campo del pensamiento

REFERENCIAS

- Batistelli, M. (2022, 29 de julio). Pensamiento computacional - SmartTEAM. SmartTEAM. <https://smartteamdigital.com/proyecto-educativo/pensamiento-computacional/>
- Bordignon, F. R. A., & Iglesias, A. (2020). Introducción al pensamiento computacional. <https://libros.unlp.edu.ar/index.php/unlp/catalog/book/1300>
- Del Angel, J. (2023, 28 de julio). ¡Desarrolla el poder del pensamiento computacional gratis con este curso en línea! Facialix. <https://facialix.com/desarrolla-el-poder-del-pensamiento-computacional-gratis-con-este-curso-en-linea>
- Flores, A. (2011). Desarrollo del pensamiento computacional en la formación en matemática discreta. Lankpsakos, No. 5. <http://www.funlam.edu.co/revistas/index.php/lampsakos/article/view/815/783>
- Jiménez, J. (s. f.). Reconocimiento de patrones con topologías diluidas. https://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0798-40652009000400004
- TecnoSalva. (2020, 14 de mayo). Robot seguidor de líneas - Tecnosalva. Tecnosalva. <https://www.tecnosalva.com/robot-seguidor-de-lineas/>
- Uscudum, E. C. L. C. Y. G. (s. f.). Descomposición | Pensamiento computacional + Microbit. <https://rea.ceibal.edu.uy/elp/pensamiento-computacional-microbit/descomposicin.html>

Glosario

Pensamiento computacional: El pensamiento computacional es un enfoque para resolver problemas que implica descomponer un problema en partes manejables, reconocer patrones, abstraer conceptos y desarrollar algoritmos o procedimientos para llegar a una solución. Es una habilidad clave en la programación y en la ciencia de la computación, que permite a los individuos formular problemas y sus soluciones de una manera que una computadora pueda ejecutar (Wing, 2006).

Diagrama de flujo: Un diagrama de flujo es una representación gráfica de un proceso o algoritmo. Utiliza símbolos estandarizados, como rectángulos, óvalos y flechas, para mostrar las operaciones, decisiones y flujos de datos dentro de un sistema. Los diagramas de flujo son herramientas útiles para planificar, documentar y comunicar cómo funciona un proceso o un programa antes de su implementación (Gómez & Morales, 2018).



La excelencia no se improvisa

síguenos

