



# Introducción al Pensamiento Computacional

Developed by Karolína Miková  
and the FERTILE Group



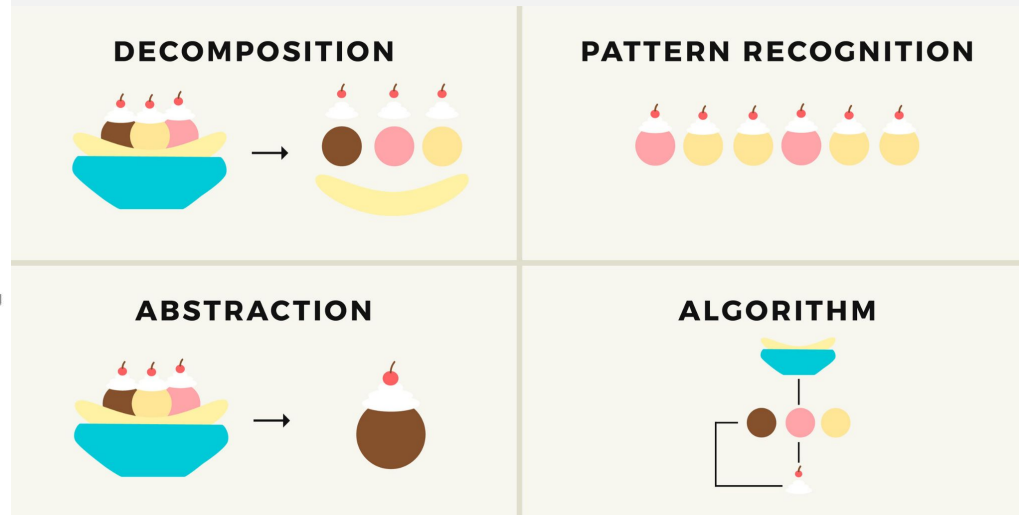
Introduction to CT © 2024 by Karolína Miková and the FERTILE group, is licensed under CC BY-SA 4.0. To view a copy of this license, visit

<http://creativecommons.org/licenses/by-sa/4.0/>



¿Qué es el pensamiento  
computacional?

# Diferentes puntos de vista



# The Computational Thinkers

## concepts



### Logic

Predicting & analysing



### Evaluation

Making judgements



### Algorithms

Making steps & rules



### Patterns

Spotting & using similarities



### Decomposition

Breaking down into parts



### Abstraction

Removing unnecessary detail



## approaches



### Tinkering

Changing things to see what happens



### Creating

Designing & making



### Debugging

Finding & fixing errors



### Persevering

Keeping going



### Collaborating

Working together

We're all computational  
thinkers here!

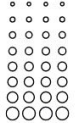
When you think about it, whether we're parents, pupils or teachers - we're all natural computer scientists, capable of computational thinking.

Our brains, like computers, process, debug and make simple algorithms every day!

## Concepts



**Decomposition**  
Dividing into parts



**Patterns**  
Finding and using  
commonalities



**Abstraction**  
Removing unwanted  
information



**Algorithms**  
Making guidelines and steps



**Logic**  
Speculating and analyzing



**Evaluation**  
Making decisions



## Approaches



**Tinkering**  
Trying new ideas



**Creating**  
Crafting and creating



**Debugging**  
Detecting and  
correcting problems



**Persevering**  
Overcome challenges  
and keep trying



**Collaborating**  
Combining efforts

## COMPUTATIONAL THINKING

Decomposition



Breaking the problem into smaller, more manageable parts.

Pattern recognition



Recognising which parts are the same and the various attributes we can use to define them.

Abstraction



Filtering out the data you need and what you don't based on the attributes.

Algorithm design



Planning the step-by-step instructions that need to be carried out to achieve the goal.

# The Computational Thinkers

# Diferentes definiciones

"Pensamiento computacional es el conjunto de procesos mentales involucrados en la formulación de un problema y la expresión de sus soluciones de tal manera que la computadora, ya sea humana o máquina, pueda llevarlas a cabo de manera efectiva." Wing, 2014

"La actividad mental para abordar problemas y formular soluciones que pueden ser automáticas." Yadav et al, 2014

"El proceso de reconocer aspectos de la computación en el mundo que nos rodea y aplicar herramientas y técnicas de la Ciencia de la Computación para comprender y razonar sobre sistemas y procesos tanto naturales como artificiales." Furber, 2012

"Una orientación mental para formular problemas como conversiones de alguna entrada a una salida y buscar algoritmos para realizar las conversiones. Hoy en día, el término se ha expandido para incluir el pensamiento con muchos niveles de abstracción, el uso de las matemáticas para desarrollar algoritmos y examinar cómo una solución se escala a través de diferentes tamaños de problemas." Denning, 2009

"Enseñar el pensamiento computacional es enseñar cómo pensar como un economista, un físico, un artista, y entender cómo usar la computación para resolver sus problemas, crear y descubrir nuevas preguntas que pueden explorarse de manera fructífera." Hemmendinger, 2010

# 5 habilidades de PC utilizadas en FERTILE

1. Abstracción
2. Reconocimiento de patrones
3. Descomposición
4. Pensamiento algorítmico
5. Evaluación

# Abstracción

1. Ocultar detalles de una idea, problema o solución que no son relevantes, para enfocarse en un número manejable de aspectos.
2. Elegir el detalle correcto para ocultar de manera que el problema se vuelva más fácil, sin perder algo importante.
3. Crear una representación (idea) de lo que estás tratando de resolver.
4. Elegir una forma de representar un artefacto para permitir que sea manipulado de maneras útiles.



# Descomposición

1. Descomponer un problema complejo en partes más pequeñas o más simples que sean más fáciles de manejar.
2. Descomponer artefactos en partes constituyentes para facilitar su manipulación.
3. Las partes pueden entenderse, resolverse, desarrollarse y evaluarse por separado.

# Reconocimiento de patrones

1. Analizar los datos y buscar patrones que den sentido a los datos o al problema.
2. Encontrar las similitudes o patrones entre problemas pequeños y descompuestos que pueden ayudarte a resolver problemas complejos de manera más eficiente.
3. Hacer predicciones sobre lo que sucederá a continuación.
4. Transferir ideas y soluciones de una área de problema a otra.

# Pensamiento algorítmico

1. Crear instrucciones paso a paso para resolver el problema o completar una tarea.
2. Expresar explícitamente los pasos del algoritmo.
3. Identificar diferentes algoritmos efectivos para un problema dado.
4. Encontrar el algoritmo más eficiente.

# Evaluación

1. Evaluar una solución a un problema y utilizar esa información nuevamente en nuevos problemas.
2. Verificar si la solución se puede generalizar mediante la automatización o extensión a otros tipos de problemas y abarcar más posibilidades/casos.
3. Evaluar si un artefacto realiza correctamente su función (corrección funcional).
4. Diseñar y ejecutar planes de prueba e interpretar los resultados (pruebas).
5. Utilizar argumentos rigurosos para verificar la usabilidad o rendimiento de un artefacto (evaluación analítica).
6. Emplear métodos que involucren la observación de un artefacto en uso para evaluar su usabilidad (evaluación empírica).



# Introducción al Pensamiento Computacional

Developed by Karolína Miková, Barbora Stenová  
and the FERTILE Group



Introduction to CT © 2024 by Karolína Miková and the FERTILE group, is  
licensed under CC BY-SA 4.0. To view a copy of this license, visit

<http://creativecommons.org/licenses/by-sa/4.0/>