Sessió 1

Marc inicial del pensament computacional

Pensament computacional i robòtica educativa · sessió 1 de 6

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Objectiu

- Entendre els quatre pilars del pensament computacional amb exemples quotidians, sense ordinador pel mig.
- Diagnosticar quins dispositius, llicències i experiència prèvia té ja el centre, abans de proposar res de nou.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Per a començar

Pregunta

Què vos ve al cap quan dic "pensament computacional"?

Quasi sempre ix "programar", "robots" o "informàtica". Al final de la sessió tornem a esta mateixa llista.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Vídeo: iniciació al pensament computacional

Vídeo

▶ Iniciación al pensamiento computacional (2026-27)

Per a veure'l, inicia la sessió amb el teu compte `@edu.gva.es` .

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Què és el pensament computacional?

Un **mètode de resolució de problemes** que utilitza la computació per a crear solucions.

I, a més, un procés que ajuda a **comprendre i resoldre dificultats sense usar ordinadors**.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

"És un pensament aplicat a la resolució de problemes i a la representació de les seues solucions, en què els resultats poden ser processats per un humà, pels ordinadors o per una combinació d'ambdós." — Jeannette Wing

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

D'on ve: 2006, Jeannette Wing

Encara que hi ha mencions anteriors, el **2006** Jeannette Wing va publicar l'article que va posar nom i marc al concepte: el "tret d'eixida" d'este moviment.

Wing el va plantejar com una **habilitat i una actitud d'aplicació universal** per a totes les persones:

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Programar per blocs: de Logo a Scratch

- **Anys 80:** el llenguatge **Logo** va portar la programació a l'escola; després va caure en desús.
- **2005-2008: Scratch, Alice, Kodu, AppInventor** i altres tornen la programació a l'aula.
- Molts usen **programació per blocs**: entorns visuals i amigables que permeten treballar el pensament computacional **des d'edats molt primerenques**.

I una idea de fons: **no cal dominar la programació de manera exhaustiva**; unes

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Quatre operacions mentals, no un aparell

No cal un ordinador per a practicar-les: amb paper, el cos mateix de l'alumnat o una tasca quotidiana n'hi ha prou.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

1. Descomposició

Dividir un problema gran en parts manejables.

Exemple: organitzar una excursió de centre no és "un bloc" — és transport, assegurança, autoritzacions, itinerari, menjar. Cada part es resol per separat.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

2. Reconeixement de patrons

Trobar semblances entre problemes per a no resoldre cada un des de zero.

Exemple: un docent que corregix exàmens reconeix els errors que es repetixen curs rere curs, i anticipa on fallarà l'alumnat.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

3. Abstracció

Quedar-se amb l'essencial d'un problema i deixar fora el detall que no aporta res.

Exemple: un plànol de metro no representa carrers ni distàncies reals, només línies i parades — l'únic que importa per a orientar-se.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

4. Algorismes

La seqüència ordenada de passos que resol un problema, escrita de manera que qualsevol que la seguisca arribe al mateix resultat.

Exemple: una recepta de cuina. Si es boten passos o es canvia l'ordre, el resultat no és el mateix plat.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Les fases, en orde

El pensament computacional s'aplica com una seqüència de passos, tots **previs al producte final** (la solució):

1. **Descompondre**
2. **Buscar patrons**
3. **Abstraure**
4. **Crear algorismes → producte final**

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Què guanyem en treballar-lo

- **Confiança** per a afrontar problemes complexos.
- Capacitat per a identificar quines parts es poden resoldre **amb mitjans informàtics**.
- **Creativitat** per a trobar solucions innovadores.
- Preparació per a aplicar l'**estratègia adequada** en cada moment.
- **Tolerància** per a acceptar solucions distintes davant d'un mateix problema.
- Habilitat per a **treballar en equip** i arribar al producte final.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

I una idea de fons: la vessant pràctica és consubstancial al mètode — **pensar, crear, provar, rectificar i tornar a provar** fins a aconseguir el resultat.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Aprendre creant: constructivisme (Seymour Papert)

Quan l'alumnat crea un producte —un programa o qualsevol altra cosa, digital o no— està **aprenent, construint, comprovant, refent i modificant**. I en analitzar els errors, és capaç de plantejar solucions noves i creatives.

És la idea de **Seymour Papert**: aprendre construint. El *constructivisme* com a aprenentatge en estat pur.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Què NO és pensament computacional

- No és sinònim de **programar** — es pot pensar de manera computacional sense escriure una línia de codi.
- No és sinònim d'**usar tecnologia** — una tauleta per a veure un vídeo no implica pensament computacional.
- No és una **assignatura a banda** — s'integra en matemàtiques, llengua, ciències o qualsevol matèria.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Per què començar sense dispositius

Idea clau

Si la primera experiència és un robot que no fa el que s'esperava, l'atenció se'n va sencera a l'aparell — i el raonament de fons es perd en la frustració tècnica.

Per això es comença amb activitats desendollades, en què l'error sempre és fàcil d'analitzar.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Relació amb el currículum

- **DigCompEdu**: dins de resolució de problemes i creació de contingut digital.
- **LOMLOE Comunitat Valenciana**: vinculat a la competència matemàtica i la competència STEM, no només a Tecnologia.
- Cabuda natural en la comprensió lectora de textos seqüencials (instruccions, procediments) i en el mètode científic.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Porta d'entrada a projectes STEM/STEAM

Molts centres volen llançar un projecte STEAM i no saben per on començar si el punt de partida és "fer un projecte gran".

Els quatre pilars, treballats primer amb activitats xicotetes i desendollades, donen confiança abans d'enfrontar-se a un projecte complex amb robòtica real.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Avaluar també el procés

No només el producte final (un dibuix, un recorregut programat, un projecte de Scratch) — també el procés:

- Corregir un error sense ajuda → validació de processos.
- Reconéixer que dos problemes es resolen igual → reconeixement de patrons.

Es reprén amb una rúbrica completa en la **sessió 6**.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Dues rutes, un producte

Essencial: rutina real de l'aula → 5–8 passos → prova literal amb una altra parella. **Repte:** corregir ambigüitats. Resultat: fitxa breu a punt per a provar.

Aprofundiment: afegir una condició i una variable. **Repte:** provar dos casos amb resultats diferents. Resultat: versió adaptable a dos nivells.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Tancament

Recapitulant

Torneu a la llista de la pissarra: què hi afegiríeu o llevaríeu ara que coneixeu els quatre pilars?

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Fonts

- **«El concepto de Pensamiento Computacional»** — INTEF, MOOC *Pensamiento Computacional educativo* (#MoocBot). D'ací ixen la definició, la cita de Jeannette Wing (2006) i el recorregut Logo → Scratch → programació per blocs.
- **«Fases del Pensamiento Computacional»** — INTEF, mateix MOOC. D'ací ixen les fases en orde, les capacitats que es desenvolupen i el constructivisme de Seymour Papert.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

Moltes gràcies!

Per a qualsevol dubte o consulta:

info@sasogu.es

Samuel Soriano · samuelsoriano.es

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.

© Samuel Soriano · Llicència CC BY-NC-SA 4.0.