



PENSAMENT COMPUTACIONAL EN LA FORMACIÓ DE MESTRES. GUIA DIDÀCTICA



Eines, reflexions, aprenentatges i exemples
pràctics resultants del projecte PECO FIM

pecofim.udg.edu

Finançat per:



“Pensament Computacional en la formació de mestres. Guia didàctica” és un document derivat del projecte PECOFIM [2015 ARMIF 00031], finançat en la convocatòria d'Ajuts de Recerca per a la Millora en la Formació Inicial de Mestres, que concedeix l'Agència de Gestió d'Ajuts Universitaris i Recerca (AGAUR).

Ha estat elaborat per **[Estebanell, M.; López, V.; Peracaula, M.; Simarro, C.; Cornellà, P.; Couso, D.; González, J.; Alsina, A.; Badillo, E.; Heras, R.]**.

Amb la col·laboració de **[Freixenet, J.; Muntaner, E.; Sabaté, F.; Serrats, L.; Córdoba, P.; Garcia, N.; Niell, M.; Bertran, A.; Bosch, D.]**.

Disseny i Maquetació: **[Ferrarons, A.]**.

Traducció: **[Weiss, M.]** (anglès), **[Bosch, D.]** (castellà).

Aquest document es distribueix sota una llicència Creative Commons Atribució-NoComercial-SenseDerivar 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-nd/4.0/>

La majoria de les imatges utilitzades en el document són pròpies o bé s'han obtingut amb llicència Creative Commons. En els altres casos, s'ha detallat les llicències corresponents al peu de la figura.

Citar com:

[Estebanell, M.; López, V.; Peracaula, M.; Simarro, C.; Cornellà, P.; Couso, D.; González, J.; Alsina, A.; Badillo, E.; Heras, R.] (2018). Pensament Computacional en la formació de mestres. Guia didàctica. Girona: Servei de Publicacions UdG.

FORMACIÓ DE PROFESSORAT SOBRE PENSAMENT COMPUTACIONAL

Eines, reflexions, aprenentatges i exemples pràctics resultants del projecte PECOFIM

El Pensament Computacional, entès com el conjunt d'estratègies i raonaments necessàries per a la resolució de problemes i el disseny de solucions sistemàtiques com ho faria una màquina o ordinador, és cada vegada més present en el món educatiu. Les activitats de programació i els robots educatius són presents, avui en dia, tant en l'ensenyament formal com en múltiples espais no formals (campus, clubs de codi, museus, etc.), i arreu afluïxen projectes educatius i les xarxes que promouen aquest tipus d'activitats. Alhora, els currículums escolars han començat a incloure la programació i la robòtica educativa com a propostes mitjançant les quals treballar la resolució de problemes, i ajudar a desenvolupar les competències digital i/o la matemàtica. Però, quines són les estratègies i capacitats necessàries que s'engloben dins del que s'anomena "Pensament Computacional"? Com s'ha de concebre aquest Pensament Computacional com a contingut escolar? I què pot fer la Universitat, responsable de la formació inicial dels mestres, per promoure la competència docent necessària en aquest àmbit?

Aquest document, resultant del projecte PECOFIM, pretén abordar aquestes diferents qüestions.

El Projecte PECOFIM (Pensament Computacional en la Formació Inicial de Mestres) ha estat finançat pel programa d'Ajuts de recerca per a la millora en la formació inicial de mestres (ARMIF), modalitat 2, de l'Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) -resolució de concessió de 29 de juny de 2016, publicada el 30 de juny de 2016-. En aquest projecte hi ha participat professorat universitari de la Universitat de Girona i de la Universitat Autònoma de Barcelona, així com mestres de diferents escoles catalanes amb una àmplia experiència en la realització d'activitats relacionades amb la programació i la robòtica en contextos educatius.

Aquest document comparteix un marc didàctic entorn al Pensament Computacional i el paper que pot tenir a l'escola com a objecte d'ensenyament i aprenentatge. A partir de recursos i exemples pràctics, s'espera contribuir a la formació del professorat sobre el pensament computacional, tot abordant el què, el com i el per a què del Pensament Computacional a l'escola.



El Pensament
Computacional en la
Formació inicial de Mestres

Web del projecte: pecofim.udg.edu

Què trobareu en aquest document?

1. De què es parla fent referència al Pensament Computacional?	6
2. Quin sentit té el Pensament Computacional a l'escola?	7
2.1. Més enllà dels conceptes computacionals: El Pensament Computacional entès com un conjunt de pràctiques i perspectives	7
2.2. Més enllà del suport digital: el Pensament Computacional com una mirada a la resolució de problemes quotidians amb i sense ordinadors	8
2.3. Més enllà de fer informàtica: el Pensament Computacional entès com una peça clau en el paradigma emergent STE(A)M	9
3. Quins són els aspectes del Pensament Computacional que cal promoure i avaluar?	10
3.1. Procediments característics del Pensament Computacional	13
3.2. Estratègies per resoldre problemes computacionals	18
3.3. Habilitats transversals en contextos computacionals	23
4. Com abordar el Pensament Computacional en la formació de mestres?	28

1. De què es parla fent referència al Pensament Computacional?

L'any 1980 Seymour Papert va introduir el terme "Pensament Computacional" en el seu llibre *"Mindstorms: Children, computers, and powerful ideas"* (traduït a l'espanyol com *"Desafío a la mente"*). El 1996, el mateix Papert, va fer una exposició de la idea en un context d'aprenentatge de les matemàtiques en el seu article *"An exploration in the space of mathematics educations"*, però no va ser fins l'any 2006 que Jeannette M. Wing va popularitzar el concepte de Pensament Computacional en l'àmbit de la investigació educativa i psicològica, publicant un article a la revista *"Communications of the ACM"* (2006). En concret, Wing va argumentar que

"el Pensament computacional implica la resolució de problemes, el disseny de sistemes i la comprensió de la conducta humana, fent ús dels conceptes fonamentals en la informàtica." (Wing, 2006: p 33).

L'autora proposa que aquesta forma de pensar és aplicable a la resolució de varis problemes, essent una habilitat fonamental per a tota la població i no només per als informàtics i programadors. Des d'aquesta perspectiva, l'autora destacava la necessitat d'integrar les idees computacionals en altres disciplines, plantejant solucions que puguin ésser portades a terme també pels humans, i no només per les màquines.

En els deu anys que han seguit després de la primera publicació de Wing, molts autors han centrat la seva atenció en la idea del Pensament computacional aportant definicions complementàries. Berry (2014) i Selby & Woollard (2014) han proposat les seves pròpies definicions, que malgrat els matisos, coincideixen en entendre el Pensament Computacional com la capacitat per identificar problemes que puguin ser resolts d'una manera similar a com ho faria un programador al donar instruccions a un ordinador fent servir un llenguatge de programació:

- dividir problemes complexos en mòduls de mida inferior,
- seqüenciar processos llargs i complexos en "passos",
- organitzar i analitzar les dades reconeixent patrons lògics,
- partir de casos concrets per arribar a situacions abstractes i generalitzables,
- utilitzar algorismes per automatitzar solucions i
- avaluar la validesa de les solucions.

PER SABER-NE MÉS

- Berry, M. (2014). *Computational Thinking in Primary Schools*. <http://milesberry.net/2014/03/computational-thinking-in-primary-schools/>
- Selby, C. & Woollard, J. (2014) *Refining and understanding Computational Thinking*. <https://eprints.soton.ac.uk/372410/1/372410UnderstdCT.pdf>
- Wing, J. (2006). *Computational Thinking*. COMMUNICATIONS OF THE ACM. Vol. 49, No. 3. <https://dl.acm.org/citation.cfm?id=1118215>

2. Quin sentit té el Pensament Computacional a l'escola?

La tecnologia informàtica, molt vinculada al pensament computacional, no és en sí mateixa una novetat a l'escola. En molts centres educatius fa moltes dècades que hi ha l' "aula d'informàtica" integrada dins l'ecosistema escolar. I, que malgrat ha experimentat canvis associats al propi desenvolupament tecnològic, tant en dispositius com en programari, sovint s'ha associat al treball amb l'ordinador (a vegades com a assignatura pròpia, a vegades integrada en altres àrees de coneixement).

Al nostre entendre, la perspectiva del Pensament Computacional pretén transcendir aquesta visió tradicional de l'aula d'informàtica delimitada en l'espai i en el temps, entenent l'activitat escolar vinculada a la programació i la robòtica com una estratègia educativa per al desenvolupament d'una competència del s.XXI, que no ha de ser una habilitat exclusiva dels professionals vinculats a carreres STEM (Ciència, Tecnologia, Enginyeria i Matemàtica), sinó per a tota la ciutadania. En aquest sentit, es presenta el Pensament Computacional a l'escola partint de tres premisses:

- Més enllà dels conceptes computacionals: El Pensament Computacional entès com un conjunt de pràctiques i perspectives.
- Més enllà del suport digital: El Pensament Computacional entès com una metodologia per a la resolució de problemes quotidians amb i sense ordinador.
- Més enllà de fer informàtica: el Pensament Computacional entès com una peça clau en el paradigma emergent STE(A)M.

2.1. Més enllà dels conceptes computacionals: El Pensament Computacional entès com un conjunt de pràctiques i perspectives

Si bé sovint s'associa el Pensament Computacional com un conjunt de continguts conceptuals a ensenyar (els conceptes clau per aprendre a programar, com són els conceptes de seqüències, bucles, paral·lelismes, esdeveniments, condicionals, operadors i dades), diferents propostes han remarcat també altres dimensions del Pensament Computacional, associades als continguts procedimentals i epistemològics. D'una banda, parlem de pràctiques computacionals com aquelles activitats pròpies del treball en contextos i amb suports computacionals, com ho són incrementar i iterar els processos, testejar i depurar els programes elaborats, reutilitzar i combinar programes, abstraure i modularitzar, entre d'altres. Finalment, també parlem de perspectives computacionals com el conjunt d'actituds i mirades a la pròpia disciplina, que inclouen aspectes com expressar, connectar, qüestionar i sentir empoderament per crear.

PER SABER-NE MÉS

- Brennan, K. & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. *Proceedings of the American Educational Research Association 2012*, pp, 1-25. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Stephenson, C., & Barr, V. (2011). *Defining Computational Thinking for K-12*. *CSTA:- The Voice of K-12 Computer Science and Its Educators*, 7(2).

Les dimensions del Pensament computacional segons Brennan i Resnick (2012)

Conceptes

- Seqüències
- Bucles
- Paral·lelisme
- Esdeveniments
- Condicionals
- Operadors
- Dades

Pràctiques

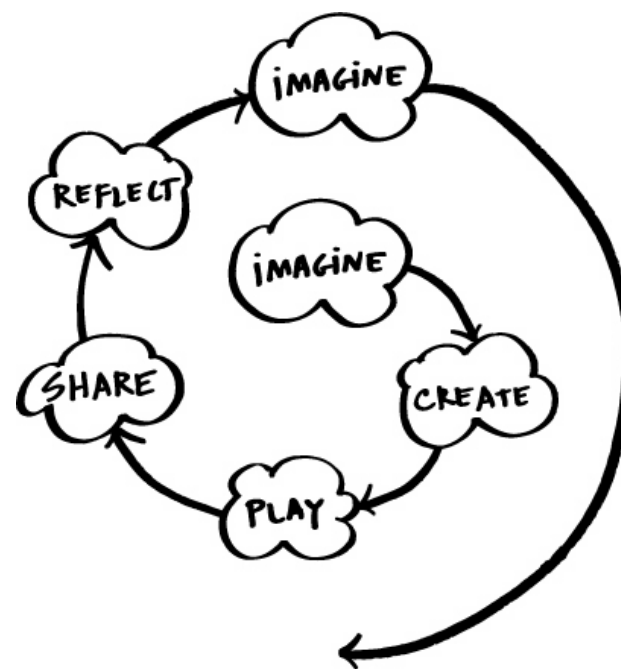
- Descompondre problemes
- Fer canvis incrementals
- Testejar i corregir
- Reutilitzar
- Persistir

Perspectives

- Expressar: Nous medis per crear
- Connectar: Poder crear amb els altres
- Qüestionar: Poder utilitzar la computació per plantejar-se preguntes i resoldre-les.

2.2. Més enllà del suport digital: el Pensament Computacional com una mirada a la resolució de problemes quotidians amb i sense ordinadors

Malgrat s'associï el Pensament computacional a l'ús de tecnologies digitals, proposem entendre'l com una manera de pensar, fer i comunicar-se que transcendeix el suport digital, i que té múltiples aplicacions en la resolució de problemes de tota mena, especialment quotidians. Així, Beauchamp (2016) destaca l'estreta relació entre el Pensament Computacional i l'enfocament educatiu de "l'Aprentatge Basat en Problemes", ja que tota resolució de problemes computa amb diferents fases: entendre el problema, plantejar un disseny, executar-lo i revisar el resultat. Per altra banda, Resnick (2007) ha defensat l'estreta relació



Espiral del pensament creatiu segons M. Resnick (2017).

<http://web.media.mit.edu/~mres/papers/CC2007-handout.pdf>

entre el Pensament Computacional i el Pensament creatiu fent servir el terme “creative computing” i destaca com els infants aprenen en l’etapa d’educació infantil, on sovint es dona un procés d’aprenentatge amb un denominador comú: imaginar, crear, jugar (experimentar), compartir, reflexionar i tornar a imaginar. Segons Resnick una de les millors maneres que tenim d’ajudar als nens i als joves, és estant segurs que tinguin l’oportunitat de seguir els seus interessos, d’explorar les seves idees i de disposar d’eines per fer sentir la seva veu. No es tracta simplement d’ensenyar-los a utilitzar la tecnologia, sinó d’oferir-los oportunitats d’utilitzar la tecnologia per crear coses.

PER SABER-NE MÉS

- Beauchamp, G. (2016). *ICT and computing in the primary school. In Computing and ICT in the Primary School: From pedagogy to practice* (p. 252).
- Griffin et al. (2012). *Assessment and Teaching of 21st Century Skills*
<http://www.springer.com/us/book/9789400723238>
- Resnick, M. (2007). *All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, 1-6.

2.3. Més enllà de fer informàtica: el Pensament Computacional entès com una peça clau en el paradigma emergent STE(A)M

La idea de Pensament Computacional ha anat agafant un creixent interès en els darrers anys en paral·lel a l’aparició del paradigma STEM (Ciència, Tecnologia, Enginyeria i Matemàtica), que sorgeix tant arrel de la creixent demanda de professionals amb perfils de caire científic, tecnològic, enginyeril i matemàtic, com arrel de noves propostes educatives que advoquen per a una major interrelació entre aquestes disciplines al llarg de l’escolarització. En paral·lel, ens trobem amb l’eclosió d’espais d’educació Maker, així com una aposta per promoure l’àrea STEAM, on les Arts tenen un paper fonamental, i transformador, que permeten diversificar tasques i feines que caldrà desenvolupar, essent complementaries i ajudant a expressar-se, connectar i comprendre el saber objecte d’estudi.

A l’hora d’entendre l’encaix del Pensament Computacional en l’àmbit educatiu STEAM, volem destacar tant les oportunitats que ofereix el Pensament Computacional com a eina clau en l’àmbit STEAM com les oportunitats que ofereix l’àmbit STEAM com a context per desenvolupar el Pensament Computacional.

Els llenguatges i suports computacionals ofereixen noves oportunitats d’enfrontar als estudiants amb reptes de caràcter científic, d’enginyeria i matemàtic, com ara modelitzar computacionalment, simular situacions del món real en entorns virtuals, representar abstraccions matemàtiques, etc. De fet, un dels principals documents referents actualment en l’àmbit STEAM, el K-12 Next Generation Science Standards (NRC, 2012), inclou l’ús del pensament computacional com una de les vuit pràctiques STEM clau. A més, desenvolupar el Pensament Computacional no només pot ajudar als estudiants a aprendre més continguts STEAM, sinó a entendre millor el paper de la computació en l’àmbit professional STEAM i l’impacte que hi té la computació. Per exemple Weintrop (2016) destaca com la biologia computacional, basada en estructures de dades i algorismes, està transformant la pròpia manera de pensar de la biologia com a ciència.

Ahora, però, l’aprenentatge en l’àmbit STEAM no només pot enriquir-se del desenvolupament del Pensament Computacional dels estudiants, sinó també a l’inrevés. Els contextos propis de l’àmbit STEAM (construcció d’explicacions

científiques i models matemàtiques, disseny de solucions d'enginyeria, etc.), són especialment idonis per promoure i donar sentit a l'ús de llenguatges computacionals, i aporten fenòmens rellevants els quals abordar des de la perspectiva computacional. I, de fet, han estat sovint les disciplines STEAM les que han aportat formes de pensar i raonar de les quals es nodreix el Pensament Computacional, com ara el pensament lògic-matemàtic o les habilitats indagatives.



Exemple de camps de recerca científica emergents basades en el suport computacional.

PER SABER-NE MÉS

- NRC. (2012). A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas. Washington, DC.: National Academy of Sciences.
- Computer Science Teachers Association: Computational Thinking resources. <http://www.csteachers.org/page/CompThinking>
- ISTE's Computational Thinking Toolkit. <https://www.iste.org/explore/articleDetail?articleid=152>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147.

3. Quins són els aspectes del Pensament Computacional que cal promoure i avaluar?

Si bé no existeix una única definició dels elements centrals o aspectes del Pensament Computacional, en la literatura especialitzada en la literatura especialitzada es poden trobar diferents propostes que destrien quins són els seus aspectes claus.

Wing, 2006	Stephenson and Barr, 2011	Selby and Woollard, 2013	Selby and Woollard, 2014
• Pensament recursiu i processament paral·lel. • Reformular un problema que inicialment sembla complicat en un problema que sapiguem solucionar. • Simplificació, integració, transformació, simulació. • Triar una representació adequada o modelar els aspectes més rellevants d'un problema per tal de fer-lo manejable. • Interpretar els codis com a dades i les dades com a codis. • Utilitzar l'abstracció i la descomposició per abordar tasques amples i complexes. • Jutjar un disseny en funció de la seva simplicitat i elegància.	• Formular problemes de manera que ens permeti utilitzar un ordinador i altres estris per solucionar-los. • Organitzar de forma lògica i analitzar dades. • Representar les dades a través d'abstraccions com són els models i les simulacions. • Generar solucions automàtiques a través de pensament algorítmic (una sèrie de passos ordenats). • Identificar, analitzar i implementar possibles solucions amb l'objectiu d'aconseguir la combinació més eficient i efectiva de passos i recursos. • Generalitzar i transferir el procés de solució d'un problema a una àmplia varietat de problemes.	• L'habilitat de pensar de manera abstracta. • L'habilitat de pensar en termes de descomposició. • L'habilitat de pensar algorítmicament. • L'habilitat de pensar en termes d'avaluació. • L'habilitat de pensar en generalitzacions.	• Un procés mental. • Abstracció. • Descomposició. • Raonament heurístic. • Pensament lògic. • Pensament matemàtic. • Pensament enfocat a l'enginyeria. • Disseny algorítmic. • Solucionar problemes. • Anàlisi. • Avaluació. • Generalització. • Recursió. • Disseny de sistemes. • Automatització. • Modelització, simulació i visualització.

Aspectes clau del Pensament Computacional

Basant-nos en aquestes diferents propostes, presentem un llistat d'aspectes clau a promoure i que podrien ser avaluable per ajudar a desenvolupar el PC a l'escola:

- Procediments característics dels llenguatges computacionals
- Estratègies per resoldre problemes computacionals
- Habilitats transversals en contextos computacionals

Procediments característics dels llenguatges computacionals	Estratègies per resoldre problemes computacionals	Habilitats transversals en contextos computacionals
 TRACTAR DADES I VARIABLES COMPUTACIONALS	 IDENTIFICAR / DELIMITAR	 CREATIVITAT I ENGINY
 REPRESENTAR ABSTRACCIONS COMPUTACIONALS	 CONSIDERAR MULTIPLES VIES	 TREBALL EN EQUIP
 AUTOMATITZAR AMB ALGORITMES COMPUTACIONALS	 DESGLOSSAR I SIMPLIFICAR	 AUTONOMIA / INICIATIVA
 SEQÜENCIAR PASSOS COMPUTACIONALS	 TESTEJAR, VALIDAR, DEPURAR	 INTERACCIÓ I COMUNICACIÓ

PER SABER-NE MÉS

- Wing, J. (2006). Computational Thinking. COMMUNICATIONS OF THE ACM. Vol. 49, No. 3. <https://dl.acm.org/citation.cfm?doid=1118178.1118215>
- Stephenson, C., & Barr, V. (2011). Defining Computational Thinking for K-12. CSTA:-The Voice of K-12 Computer Science and Its Educators, 7(2).
- Selby, C. & Woollard, J. (2013) Computational Thinking: The Developing Definition. <https://eprints.soton.ac.uk/356481/>
- Selby, C. & Woollard, J. (2014) Refining and understanding Computational Thinking. <https://eprints.soton.ac.uk/372410/1/372410UnderstdCT.pdf>

3.1. Procediments característics del pensament computacional

Un dels aspectes clau del PC és la utilització dels llenguatges computacionals. Si bé existeix una gran varietat de llenguatges (llenguatges de codi, per blocs, amb icones, mixtes, etc.), existeixen alguns denominadors comuns.

Un d'aquests denominadors és el tractament de **dades i variables computacionals**, tot entenent d'on provenen, quin és el seu significat i com poden fer-se servir per generar noves dades, així com reconèixer patrons per organitzar aquestes dades. Els reptes computacionals als que s'enfronten els estudiants a l'escola sovint requereixen un treball amb dades numèriques i d'un tractament sistemàtic d'aquestes, la identificació de les variables que permeten descriure i controlar un procés computacional, i la relació entre aquestes variables.

Per això també diem que requereix desenvolupar la capacitat de **representar abstraccions computacionals** de tota mena: condicions, patrons, relacions geomètriques, relacions algebraïques, etc., que s'expressen a través de codi computacional.

De fet, abans de programar amb codi, és convenient que els estudiants representin les seves idees amb paper i llapis, així com a través de jocs vivencials, per posteriorment traduir-ho al llenguatge computacional.

A més, una de les maneres més comuns per representar i estructurar les relacions entre variables i les condicions que les determinen és a través dels **algoritmes computacionals**, que permet automatitzar el funcionament d'un programa informàtic o el comportament d'un robot, fent servir bucles, condicionals, operadors, variables i paràmetres formant un algoritme per resoldre casos particulars d'un problema general.

Alhora, això passa per desenvolupar habilitats entorn a **seqüenciar passos computacionals**, sabent ordenar de forma òptima les instruccions que configuren un programa, tot fent servir seqüències, repeticions o paral·lelismes.



TRACTAR DADES I
VARIABLES
COMPUTACIONALS



REPRESENTAR
ABSTRACCIONS
COMPUTACIONAL



AUTOMATITZAR
AMB ALGORITMES
COMPUTACIONALS



SEQÜENCIAR
PASSOS
COMPUTACIONALS



TRACTAR DADES I VARIABLES COMPUTACIONALS

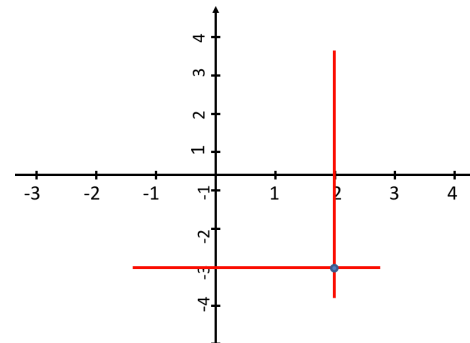
EXEMPLE PRÀCTIC

Triem quines coordenades ens ajuden millor a definir un moviment en el pla

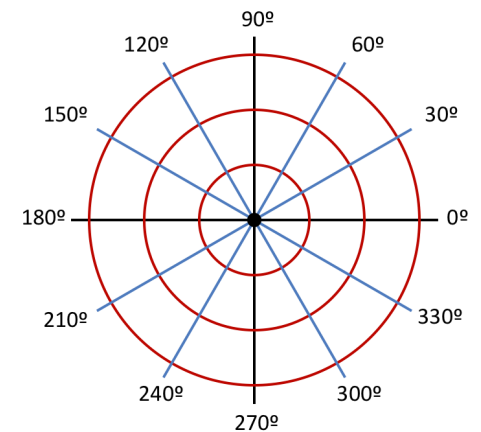
Quan es proposa a un grup d'infants dissenyar un petit videojoc amb Scratch, un dels principals reptes que es troben és el de definir el moviment dels personatges. Aquest llenguatge permet definir aquest moviment tant a partir de les coordenades cartesianes (X i Y) com de les polars (direcció i desplaçament), i segons com vulguin interactuar amb els personatges, els infants hauran de triar quin tipus de coordenades es poden fer servir com a dades en el seu codi.

Per exemple, si es vol definir el moviment d'un personatge que xoca contra els extrems de la pantalla, la dada que els serà més útil és la direcció expressada en graus, mentre que si volen definir la caiguda vertical d'un personatge, la dada que els serà més útil és la coordenada Y.

Eixos cartesianes



Coordenades polars





REPRESENTAR ABSTRACCIONS COMPUTACIONALS

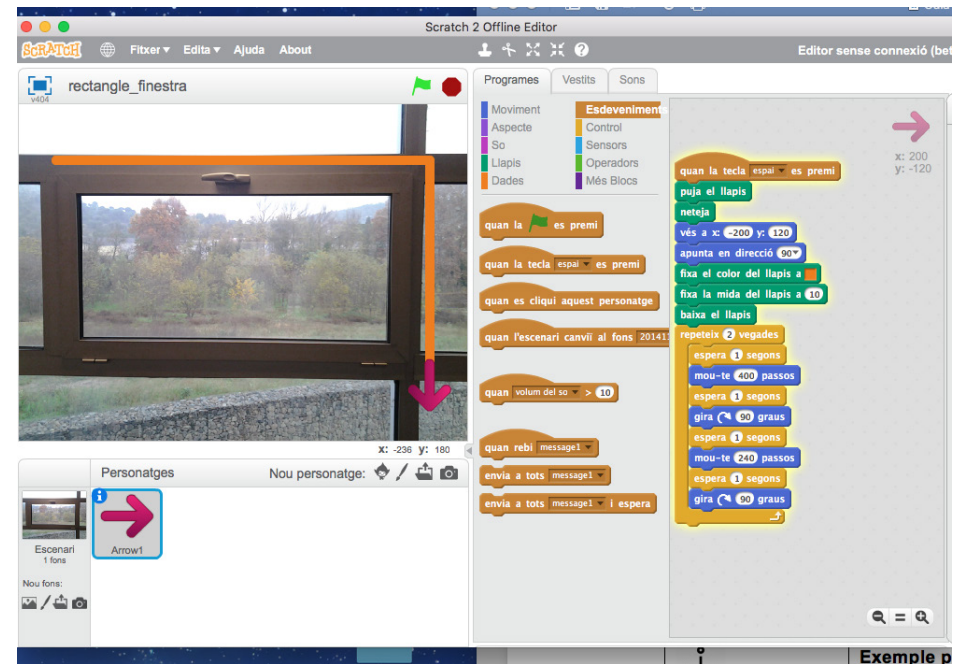
EXEMPLE PRÀCTIC

Reproduïm els polígons que veiem al nostre voltant

Podem proposar als infants reconèixer formes geomètriques planes en el seu entorn físic, per tal que les fotografiïn i les arribin a descriure matemàticament amb l'ajuda de l'entorn de programació Scratch.

Per fer-ho, poden col·locar una de les fotografies com a escenari, i programar un personatge per a què, amb un clic, ressegueixi el contorn de la forma que hi apareix.

Això requereix un procés d'abstracció on hi entren en joc dades i variables com la llargada dels costats del polígon i l'angle que ha de girar el personatge en cada vèrtex.



Exemple p



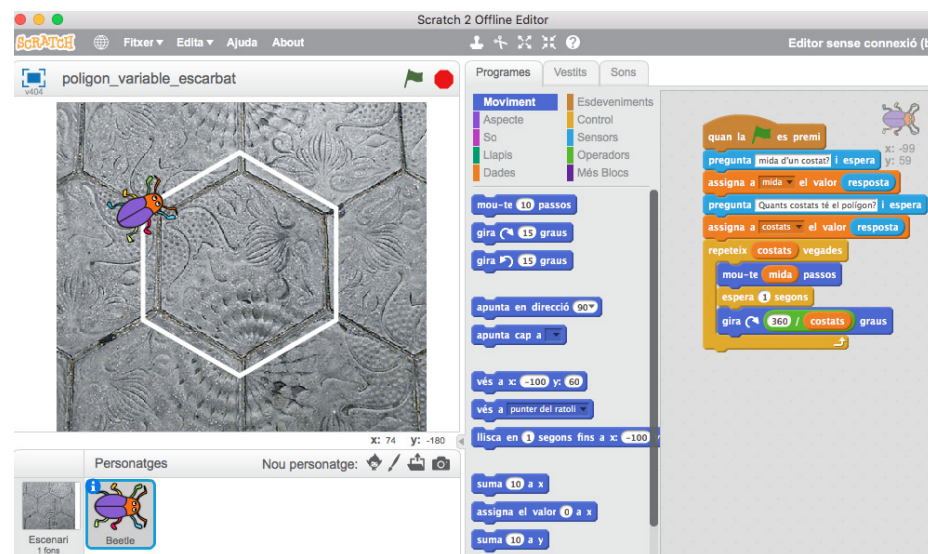
AUTOMATITZAR AMB ALGORITMES COMPUTACIONALS

EXEMPLE PRÀCTIC

Polígons regulars: com podem generalitzar la seva expressió matemàtica?

Continuant amb l'exemple anterior, en el cas de que la forma sigui un polígon regular, els nens poden arribar a trobar la relació entre l'angle de gir i el nombre de costats del polígon, relacionant els dos paràmetres amb un operador, que en aquest cas serà la divisió.

Podran, així, construir un algoritme general per dibuixar el contorn de qualsevol polígon regular a través de la relació que estableix que cada angle de gir és 360 dividit pel nombre de costats del polígon.





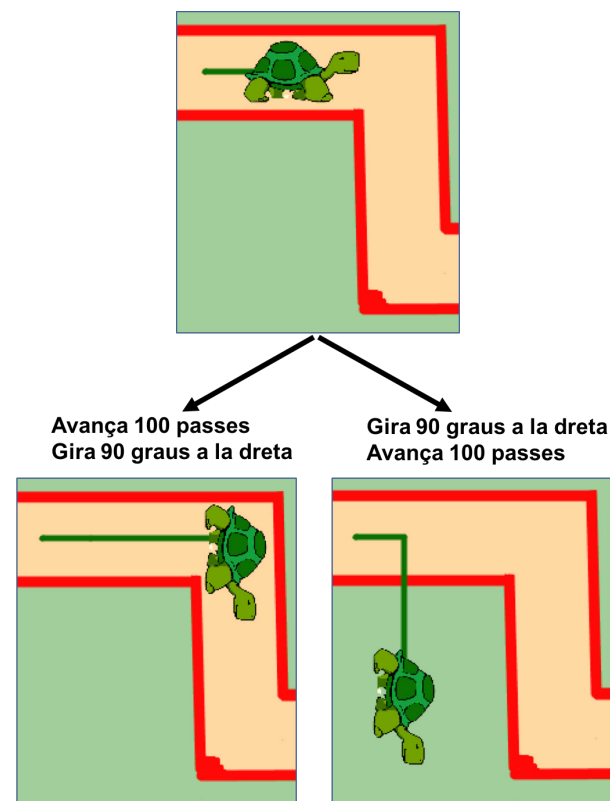
SEQUENCIAR PASSOS COMPUTACIONALS

EXEMPLE PRÀCTIC

Descobrim que les instruccions que es donen a un robot no compleixen la propietat commutativa

Si es proposa a un grup d'infants programar un robot, o un personatge que hagi de seguir un itinerari concret, aquests hauran de definir una seqüència d'ordres o instruccions que aquest personatge haurà de seguir.

Si les dues instruccions de que es disposa són "Avançar 100 passos" (moviment de translació) i "Girar 90° a la dreta" (moviment de rotació), els infants poden comprovar que l'ordre amb que es col·loca la seqüència d'instruccions no porta al mateix resultat, i que, per tant, les instruccions donades al robot no compleixen la propietat commutativa.



3.2. Estratègies per resoldre problemes computacionals

En paral·lel al domini dels elements clau del domini dels llenguatges computacionals (els seus components, les seves estructures, regles, etc.), el PC té una estreta relació amb la manera com les persones ens enfrontem a la resolució de problemes.

Les estratègies de resolució de problemes, de fet, han estat àmpliament discutides en la literatura educativa sota diferents paraigües (habilitats de resolució de problemes, pensament crític, etc). Ara bé, les habilitats per desenvolupar aquestes estratègies no són universals i deslligades dels contextos en que s'apliquen, i, per tant, prenen un sentit molt particular quan es tracta d'enfrontar-se i resoldre problemes computacionals.

Així, el pensament computacional implica la capacitat per **identificar i delimitar** un problema computacional a resoldre, tot analitzant les seves possibles solucions, optimitzant els passos i recursos. Quan es fa, cal també **contemplar**

múltiples vies, entenent que hi pot haver diversos camins o solucions vàlides. La resolució de problemes passa també per la capacitat de desglossar i simplificar, tot modularitzant i dividint situacions complexes en altres més senzilles.

Finalment, també cal **testejar, validar i depurar** les solucions de forma iterativa, tot aprenent dels errors identificats, ja que en computació, depurar implica cercar l'error entenent què és el que no funciona.



IDENTIFICAR I DELIMITAR



CONSIDERAR
MULTIPLES VIES



DESGLOSSAR I
SIMPLIFICAR



TESTEJAR, VALIDAR,
DEPURAR



IDENTIFICAR I DELIMITAR

EXEMPLE PRÀCTIC

Ens preguntem què tenim i què ens cal abans de començar a programar un robot educatiu programable (Bee-Bot, Cubeto, altres)

Quan un grup d'infants es troba davant el repte de programar un robot que ha de desplaçar-se des d'un lloc a un altre, és convenient que prèviament reflexionin sobre quin és el problema al qual s'enfronten.

Per exemple, no és el mateix definir un desplaçament a partir d'un punt de sortida i un punt d'arribada que arribar a un punt partint d'un de sortida i havent fet un desplaçament. És necessari, per tant, delimitar quin és exactament el repte que han d'abordar, i identificar quines possibles maneres hi ha de resoldre'l.





CONSIDERAR MÚLTIPLES VIES

EXEMPLE PRÀCTIC

Construïm una torre amb gots (I)

Una activitat comuna, que permet reflexionar sobre la necessitat del llenguatge computacional per donar instruccions clares a un robot, és emular la programació d'un braç robòtic que ha de construir una torre de gots amb una estructura donada.

Per equips, alguns infants o joves poden fer de programadors, havent d'escriure el programa que permetrà al braç robòtic construir l'estructura, utilitzant un conjunt de símbols prèviament pactat (que indiquin moure's a dreta, esquerra, amunt, avall, girar el got, etc.). Un cop hagin escrit el programa, un altre grup d'infants o joves que no ha participat en l'escriptura, executarà les instruccions construint la torre de manera real amb gots físics.

Així, els programadors discutiran i reflexionaran sobre les diferents maneres que hi ha de construir la torre, en quin ordre col·locar els diferents gots i els avantatges de cada opció.





Construïm una torre amb gots (II)

Una possible activitat per treballar el “depurar”, podria ser la de proveir un exemple de codi que no fa el que hauria de fer, i fer analitzar aquest codi als infants per tal de cercar l’error, fent així un exercici que implica entendre perquè no funciona un programa i corregir-lo per aconseguir una possible solució.

ns:

		
Agafa el got i puja el braç	Baixa el braç i deixa el got	Puja el braç
		
Avança mig got	Retrocedeix mig got	Inverteix el got

$d \rightarrow b$
 $d \rightarrow \rightarrow b$
 $d \rightarrow \rightarrow \rightarrow b$
 $d \rightarrow \rightarrow \rightarrow \rightarrow b$
 $d \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow b$

$P = 0$
 Repeteix 5 vegades

$$\left\{ \begin{array}{l} P = P + 2 \\ \text{d} \uparrow P (\rightarrow) \text{ c} \downarrow \text{b} \uparrow P (\leftarrow) \end{array} \right.$$



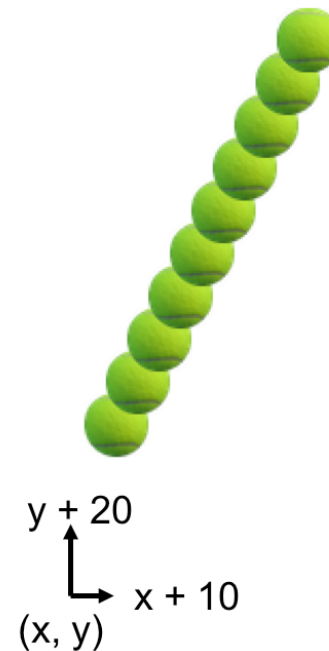
DESGLOSSAR I SIMPLIFICAR

EXEMPLE PRÀCTIC

Descomponem un moviment complex en dos de més simples

Sovint els infants volen reproduir computacionalment comportaments o fenòmens complexos, que requereixen, prèviament, d'una simplificació a partir de traduir un repte gran en varis de petits.

Pensem, per exemple, en la composició dels moviments. Un moviment complex pot estar compost per varis de més simples, com ara un moviment diagonal pot estar compost per un de vertical i un d'horitzontal.



**Repeteix 10 vegades:
llisca en 1 segon**

fins a $x = x + 10$

fins a $y = y + 20$

3.3. Habilitats transversals en contextos computacionals

En els darrers anys s'han fet múltiples propostes de formular les competències transversals i/o les competències del s.XXI, que engloben nocions clàssiques de HOTS (High Order Thinking Skills), les també anomenades soft skills, i altres aspectes per tenir en compte les demandes de la dinàmica, global i profundament digital de la societat actual. Igual com succeeix amb la resolució de problemes, la literatura cognitiva sembla indicar que tota competència es desenvolupa i s'expressa d'una forma determinada segons el context, i per tant, la qüestió no és tant quines són les habilitats transversals en abstracte o desvinculades dels continguts escolars, sinó quin sentit prenen en contextos computacionals com els que es discuteixen en aquest document.

Parlem d'habilitats transversals com la **creativitat, el treball en equip, l'autonomia i la iniciativa personal o la interacció i comunicació**; no com a competències transversals a desenvolupar “a part” del pensament computacional, sinó com a part intrínseca del pensament computacional.

Dit d'una altra manera: no cal posar el focus en ensenyar la creativitat o el treball en equip, sinó el que cal és poder aconseguir ensenyar computació desenvolupant la creativitat i el treball en equip.



CREATIVITAT I ENGINY



TREBALL EN EQUIP



AUTONOMIA I INICIATIVA



INTERACCIÓ I
COMUNICACIÓ



CREATIVITAT I ENGINY

EXEMPLE PRÀCTIC

Pensem solucions enginyoses per fer que el robot segueixi una línia

Quan un grup d'infants disposen d'un robot amb rodes i un sensor de llum, el qual han de fer guiar emprant una línia de color dibuixada al terra, existeixen diferents maneres creatives d'enfrontar-se al repte.

Per exemple, el grup d'infants pot situar el robot al marge de la línia (frontera entre fosc i clar) de manera que el robot avanci en petits trams, fent un gir cap al color fosc quan el sensor detecta claror i fent un gir cap al color clar quan el sensor detecta foscor. Per fer cada gir només es mou una de les dues rodes i s'avança, en comptes de movent les dues rodes en línia recta, movent alternativament cadascuna d'elles.

Per treballar la creativitat és important dissenyar els tallers on s'hagin d'utilitzar diferents materials, deixar que els nens expressin la seva veu, deixar que segueixin els seus interessos. Que inventin poemes, històries, robots i jocs a partir del que els agrada, i que aprenguin a reutilitzar i barrejar projectes i solucions, i expressin lliurement les seves idees.





TREBALL EN EQUIP

EXEMPLE PRÀCTIC

Creem una història multimèdia col·laborativa

El desenvolupament de programes computacionals a nivell professional requereix sovint del treball en equip, i aquest es pot reproduir de manera anàloga dins de l'aula.

Per exemple, proposant a un grup d'infants el repte de crear una història treballant de manera col·laborativa. Inicialment, l'argument general de la història és consensuat entre tota la classe, però després els infants es divideixen en grups i cadascun d'aquests grups en fa una part, realitzant-ne un guió i programant la seva animació digital amb Scratch.

Per visualitzar el resultat final es situen els ordinadors en fila, engegant el seu programa tots al mateix temps de manera que hi hagi personatges que, aparentment, saltin d'un ordinador a l'altre. La coordinació entre grups ha de permetre tant la coherència argumentativa de les parts de la història com la coherència dels programes, garantint la sincronització d'esdeveniments a partir de l'acord dels temps a transcorre des del començament de l'execució.





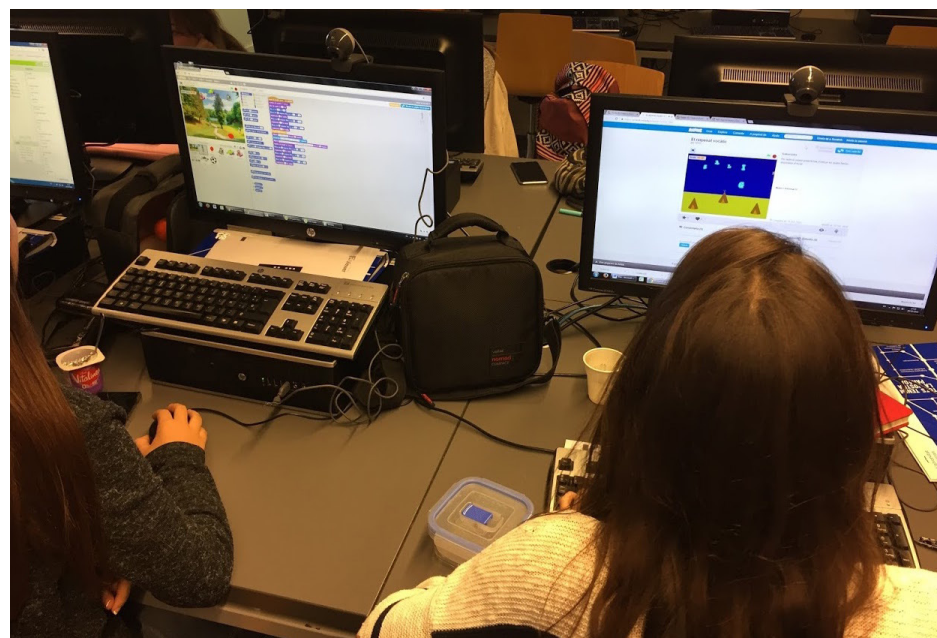
AUTONOMIA i INICIATIVA

EXEMPLE PRÀCTIC

Decidim fins a quin grau de dificultat volem arribar quan dissenyem un videojoc

Amb l'objectiu final que cada grup de treball creï un videojoc al que puguem jugar plegats, a l'aula es presenten diferents eines de creació de videojocs que van des de les més simples a les més complexes: Twine, Scratch, Kodu, Stencyl i Unity. La corba d'aprenentatge de cadascuna de les aplicacions és diferent. Això vol dir que, si un grup tria desenvolupar el seu projecte amb una eina que comporta certa complexitat, haurà de destinar una part important del seu temps a aprendre, pel seu compte, el funcionament de l'eina.

L'experiència ens diu que aproximadament un terç dels grups decideix desenvolupar el videojoc amb una eina que els exigirà un grau elevat d'autonomia per trobar solucions als problemes que se'ls aniran plantejant en el decurs del desenvolupament del seu projecte.





INTERACCIÓ I COMUNICACIÓ

EXEMPLE PRÀCTIC

Pensem com hem de donar instruccions el màxim de clares i entenedores a l'Infant-bot

Per donar valor als tallers i als projectes que es desenvolupen a l'aula en un moment determinat; un alumne/a pot explicar una troballa que hagi fet, als seus companys, o al final del taller presentar el resultat i el procés a la classe (o a l'escola), documentar els treballs, i/o en acabar un taller fer-los escriure als nens "consells"; que donarien a un company si haguessin detornat a fer el taller.

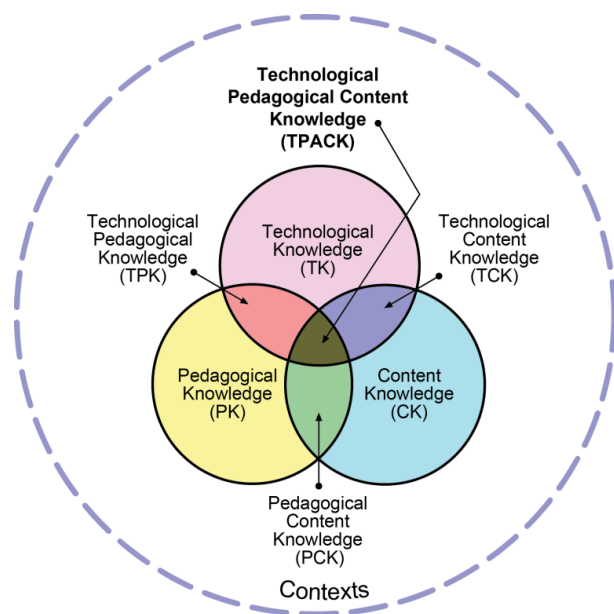
Per exemple, en una activitat anomenada "Infant-bot" (nen que executa un programa com si fos un robot), els nens poden fabricar ells mateixos els codis per donar les instruccions al nen-bot en el qual es pot anomenar *el llenguatge dels robots*, un conjunt de símbols que ells consensuaran. Els nens que el programaran han de poder comunicar-se amb els companys que faran de robot donant-los les instruccions de manera clara i entenedora.



4. Com abordar el Pensament computacional en la formació de mestres?

Degut a tractar-se d'un camp força emergent, el pensament computacional juga ara mateix un paper poc rellevant en la formació inicial de mestres que es du a terme en l'àmbit universitari. Alguns estudis han identificat aquesta mancança en la formació inicial del professorat (Bower & Falkner, 2015).

Abordar el pensament computacional en la formació inicial de mestres pot concebre's com una integració del que s'ha anomenat el coneixement didàctic-tecnològic del contingut (Shulman, 2005), normalment anomenat TPCK per les seves sigles en anglès (també conegut com a model TPACK, Technological Pedagogical and Content Knowledge). EL TPCK es pot entendre com "la capacitat del professorat per transformar el seu coneixement del contingut en formes que siguin didàcticament poderoses i tot i així adaptades a la varietat que presenten els seus alumnes en quant a habilitats i bagatges"



Model TPACK, o TPACK, Technological Pedagogical and Content Knowledge (Koehler, Mishra i Cain, 2013)

LLICÈNCIA: "Reproduced by permission of the publisher, © 2012 by tpack.org"

Igual com succeeix amb altres processos de formació del professorat, aquesta formació ha d'entendre's a diferents nivells, els quals es combinen i s'interrelacionen al llarg de la progressió d'aprenentatge que aquests futurs mestres fan al llarg de la seva formació inicial.

Si l'objectiu final de la formació de professorat en Pensament Computacional és que aquests siguin capaços de generar situacions d'aprenentatge amb els seus futurs estudiants, el primer que cal és que el professorat en formació experimenti en primera persona un aprenentatge i desenvolupament del seu propi Pensament Computacional: un coneixement de les eines que existeixen, dels llenguatges computacionals, el desenvolupament d'estratègies de raonament i resolució de problemes, etc. Això només és possible si els futurs mestres s'enfronten ells mateixos a la resolució de problemes computacionals com a estudiants/aprenents: programar un robot, dissenyar un videojoc o història interactiva, desenvolupar una app, o també resoldre problemes computacionals "unplugged". És per aquest motiu que diem que un primer nivell necessari en la formació de mestres és el **nivell usuari**. En aquest nivell formatiu no es busca que el futur mestre es plantegi encara com ajudar a desenvolupar el Pensament Computacional a una altra persona (o els seus potencials alumnes), sinó que es plantegi qüestions com ara com fer servir un llenguatge computacional determinat per assolir alguna les fites proposades (el robot, el videojoc, la app, etc.).

Només un cop una persona ha experimentat el fet d'enfrontar-se a un repte computacional dissenyant un petit programa, elaborant un codi, etc., és possible fer una reflexió sobre què ha hagut de fer. Entenem, per tant, el **nivell del usuari reflexiu** com aquell on el futur mestre està reflexionant sobre el que ha fet quan ha hagut d'enfrontar-se a un repte computacional: quins raonaments ha

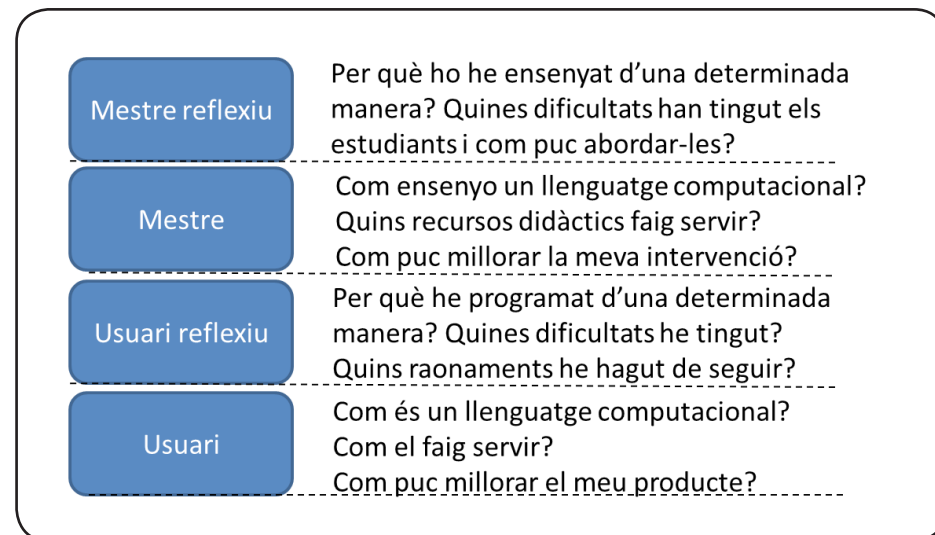
PER SABER-NE MÉS

- Bower, M. i Falkner, F. (2015) Computational Thinking, the Notional Machine, Pre-service Teachers, and Research Opportunities. Proceedings of the 17th Australasian Computing Education Conference, 27-30.
- Koehler, M. J., Mishra, P., i Cain, W. (2013). What is technological pedagogical content knowledge (TPACK)? Journal of Education, 193(3), 29-37
- Koehler, M. J., Mishra, P., i Cain, W. (2015). ¿Qué son los Saberes Tecnológicos y Pedagógicos del Contenido (TPACK)? Virtualidad, Educación y Ciencia, 10 (6), pp. 9-23. <http://www.punyamishra.com/wp-content/uploads/2016/08/11552-30402-1-SM.pdf>

fet servir? Quins elements del llenguatge computacional ha fet servir i perquè? Aquest és un exercici de metacognició en el qual prenen especial rellevància els 12 aspectes claus discutits en l'apartat anterior, ja que permeten reflexionar de forma estructurada sobre les actuacions pròpies.

Només després d'haver reflexionat com a usuari/a té sentit situar als futurs mestres davant el **repte de fer de mestres** pròpiament dit, és a dir, de decidir què vol ensenyar, què esperen que els seus estudiants aprenguin sobre pensament computacional, quins recursos i estratègies seleccionar, etc. En la formació inicial de mestres existeixen diferents moments on fer aquest procés: en els pràctics i estades a l'escola, en activitats de col·laboració universitat – escola (visites, tallers, fires, etc.), però també fent que els futurs mestres simulin petites situacions d'aula entre iguals, els anomenats “microteachings” on uns fan de mestres i els altres fan d'alumnes.

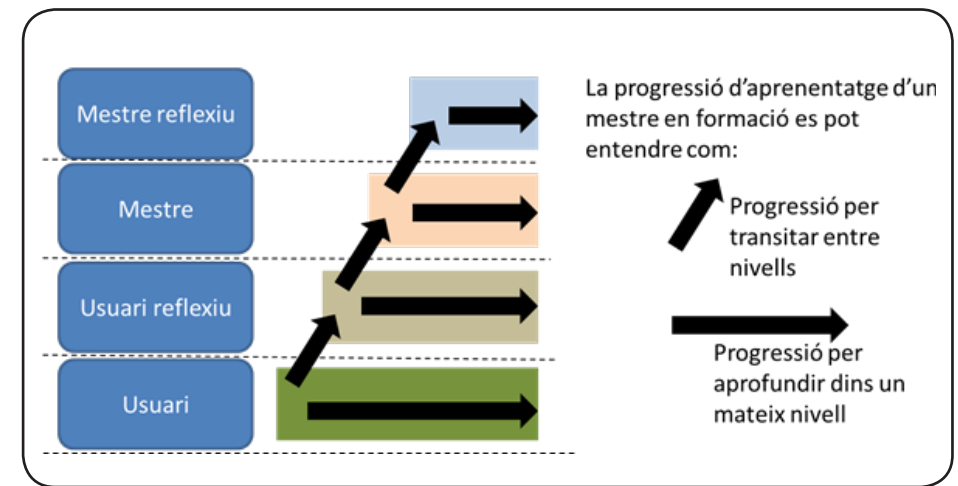
Finalment, només després d'haver experimentat en primera persona i haver fet de mestre és possible reflexionar sobre el procés d'ensenyament i aprenentatge relacionat amb el pensament computacional, és a dir, fer de **mestre reflexiu**. Això passa per no només preguntar-se què i com ensenyar, sinó per què, sobre les dificultats i casuístiques de cada context educatiu determinat i les estratègies per abordar-les.



Nivells de progressió en l'aprenentatge del Pensament Computacional dels futurs mestres.

La proposta d'aquests quatre nivells clau serveixen per orientar el que pot ser la progressió d'aprenentatge en pensament computacional dels futurs mestres. D'una banda, la progressió d'aprenentatge pot entendre's com l'aprofundiment progressiu dins d'un mateix nivell. Per exemple, en el nivell usuari, la formació de mestres pot estar centrada en aprendre llenguatges computacionals i estratègies de resolució de problemes computacionals cada cop més sofisticades. En el cas de la robòtica, un possible exemple podria ser introduir els robots Bee-bot (<https://www.bee-bot.us/>) al començament, seguir amb els robots Lego WeDo (<https://education.lego.com/>), i posteriorment continuar amb els robots Edison (<https://meettedison.com/>), i/o els mbots (<https://www.makeblock.com/steam-kits/mbot>) i/o els Lego Mindstorms (<https://education.lego.com/>)

Ahora, la progressió pot entendre's com el trànsit entre nivells, fent que els mestres en formació siguin primer usuaris, després usuaris reflexius, després mestres i finalment mestres reflexius. Així, per exemple, en una seqüència formativa seria convenient proposar-los primer fer un petit videojoc amb Scratch, després reflexionar sobre què han après, després dur a terme un taller (amb altres mestres en formació o amb estudiants) i finalment reflexionar sobre què han après a l'hora d'haver d'ensenyar-ho, quines dificultats han presentat els seus “alumnes” i com els hi podrien ajudar a superar-les i/o evitar-les.



Etales en la progressió de l'expertesa en Pensament Computacional dels mestres.

Aquesta doble manera d'entendre la progressió d'aprenentatge en la formació de mestres (esdevenint cada cop més expert/a en cada nivell i essent capaç de progressar cap el següent nivell) permet situar cada activitat formativa, i així traçar itineraris formatius que puguin ser aplicables a d'altres contextos de formació del professorat.

EXEMPLE PRÀCTIC:

Taller de disseny d'un videojoc per aprendre a programar a partir de petits reptes

NIVELL USUARI

Per tal d'aprendre i dominar el llenguatge de programació Stencyl, els futurs mestres han de crear un videojoc que es basi en el conegut videojoc "Galaga". La seqüència formativa comença amb una breu introducció sobre el funcionament bàsic de l'eina: la interfície, les regles de funcionament, les peces de codi, etc. Un cop presentat aquest llenguatge programació, es van presentant als participants petits reptes que han d'anar resolent, amb dificultat creixent. Periòdicament es fan posades en comú per tal de resoldre els reptes que han anat sorgint. Al final del procés els participants han adquirit un coneixement prou sòlid sobre programació per blocs que els permetrà tant aprofundir en el futur amb llenguatges de programació més sofisticats com ara començar a preguntar-se com aquest aprenentatge que ells han experimentat pot ser ensenyat per ells i elles en el futur.

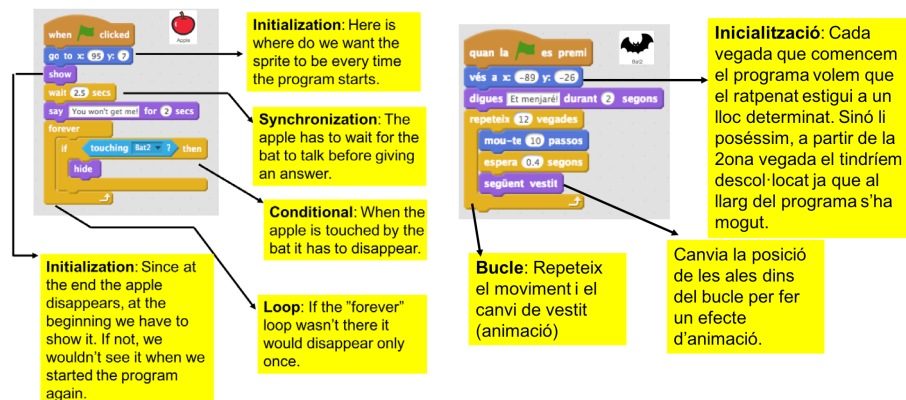
EXEMPLE PRÀCTIC:

Activitat d'anàlisi d'un programa després d'haver-lo dissenyat

NIVELL USUARI REFLEXIU

Per tal d'aprendre i dominar el llenguatge de programació Stencyl, els futurs mestres han de crear un videojoc que es basi en el conegut videojoc "Galaga". La seqüència formativa comença amb una breu introducció sobre el funciona-

ment bàsic de l'eina: la interfície, les regles de funcionament, les peces de codi, etc. Un cop presentat aquest llenguatge programació, es van presentant als participants petits reptes que han d'anar resolent, amb dificultat creixent. Periòdicament es fan posades en comú per tal de resoldre els reptes que han anat sorgint. Al final del procés els participants han adquirit un coneixement prou sòlid sobre programació per blocs que els permetrà tant aprofundir en el futur amb llenguatges de programació més sofisticats com ara començar a preguntar-se com aquest aprenentatge que ells han experimentat pot ser ensenyat per ells i elles en el futur.



EXEMPLE PRÀCTIC:

Microteaching entre iguals sobre pensament computacional

NIVELL MESTRE

Després d'haver conegut el llenguatge de programació Scratch (nivell usuari) i després haver reflexionat sobre quins aspectes del pensament computacional es treballen quan els infants creen una petita aplicació a través d'aquest llenguatge (nivell usuari reflexiu), es proposa a alguns d'estudiants del grau de mestres dissenyar i implementar davant dels seus companys una petita intervenció de 20 minuts, per tal d'ensenyar-los algun aspecte particular de Scratch. Així, els participants encarregats de fer aquesta intervenció, que anomenem microteaching, han de pensar quins són els objectius d'aprenentatge que es plantegen, quins passos seguiran, com gestionaran el treball del grup, quins exemples posaran, etc., Això permetrà, posteriorment, reflexionar sobre què ha funcionat millor i què es podria millorar de la seva intervenció.

EXEMPLE PRÀCTIC:

Valoració posterior a la realització d'activitats de Pensament Computacional en una intervenció amb infants fora de l'aula

NIVELL MESTRE REFLEXIU

En un context d'actuació amb infants reals, un grup d'estudiants en formació pot implementar propostes que han dissenyat per a ser desenvolupades amb nens i nenes, per exemple emprant algun robot o sent els mateixos infants els que actuïn reben ordres com si fossin robots,. Aquests estudiants tenien el repte no només de dur a terme l'activitat en sí, sinó anar recollint evidències de l'aprenentatge fet pels infants que hi ha participat: converses entre infants, situacions crítiques, etc. Un cop recollides aquestes evidències, els mestres en formació es troben de nou a la universitat per comentar la seva experiència. A través de les evidències recollides i d'una graella d'aspectes clau del pensament computacional (veure secció anterior) els participants discuteixen sobre quin tipus d'aprenentatges poden haver promogut entre els infants que participaven del taller, aconseguint, així, fer una reflexió sobre la seva pràctica.

PER SABER-NE MÉS

- BBC Bitesize <https://www.bbc.com/education/guides/zp92mp3/revision>
- Bocconi et al., (2016). Developing Computational Thinking in Compulsory Education. Implications for policy and practice. JCR Science for Policy Report. European Commission. http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf
- CAS Barefoot <https://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking-further-examples/>
- Computing at School resources: <https://www.computingschool.org.uk/>
- Computer Science Without a Computer: <https://csunplugged.org/en/>, <http://csedweek.org/unplugged/thinkersmith>
- Google Education Computational Thinking microsite <https://edu.google.com/resources/programs/exploring-computational-thinking/>
- Google CT for Educators https://computationalthinkingcourse.withgoogle.com/course?use_last_location=true
- Kiki Computational Thinking Games <http://games.thinkingmyself.com/>

Abans d'acabar, alguns consells pràctics:

- **No cal saber-ho tot abans de començar:** Per descomptat que una cosa a aprendre és que no cal saber-ho tot sobre una determinada eina. Si que cal afrontar un repte o problema i plantejar i desenvolupar una possible solució. Però és evident que quan aquest repte o problema es desenvolupi en una aula hi haurà múltiples idees i camins que els estudiants exploraran i, davant dels dubtes dels nens i nenes, el mestre haurà d'aprendre a respondre “explora, investiga”; a plantejar preguntes que els facin pensar, que els orientin cap a descobrir i solucionar els petits entrebancs que es puguin anar trobant.
- **Convé plantejar reptes col·laboratius:** És fonamental que els estudiants treballin en equip, i aprenguin a col·laborar, comunicar, i gestionar problemes i solucions en el marc del treball en equip.
- **Cal tenir en compte els espais a on desenvolupar les activitats:** És molt important que els mestres/futurs mestres tinguin consciència dels espais on es realitzaran els tallers/activitats. És clau per l'èxit del taller que els espais siguin adequats per treballar en equips de nens/nenes, que els espais convidin a compartir projectes i idees, i que l'equipament resulti funcional.
- **Els equipaments han d'estar a punt abans de començar l'activitat:** És important haver provat els equips i tecnologies prèviament a la sessió de treball amb els nens i les nenes, i que, quan comencin el taller es trobin els materials a punt; així estalviarem problemes relacionats amb la tecnologia (no hi ha wifi, no funciona la wifi, no hi ha bluetooth, aquest plugin no està instal·lat, no funciona amb aquest navegador,...).
- **Si malgrat tot alguna cosa no funciona tal com s'havia esperat, no passa res, l'error, i saber-lo solucionar, formen part del repte!**

