

Pensament computacional a l'aula: recursos i estratègies

Autoria: Nivard Arboix González

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats per la professora: Teresa

Romeu Fontanillas

PID_00301454

Primera edició: setembre 2024

Introducció

1. Definició i justificació

1.1. Definició

1.2. Justificació

2. Les cinc habilitats fonamentals del pensament computacional

2.1. Introducció

2.2. Algorítmica

2.3. Descomposició

2.4. Abstracció

2.5. Generalització i patrons

2.6. Avaluació

2.7. Resum d'idees clau

3. Connexions entre el pensament computacional i àmbits curriculars de primària

Conclusió

Bibliografia

Introducció

El pensament computacional és una competència que ajuda l'alumnat a desenvolupar habilitats de pensament tant en el marc de la computació com en diversos àmbits de la vida. És una competència bàsica i fonamental per als ciutadans de la societat digital que ha de formar part de l'ensenyament curricular. Per aquest motiu, és necessari que els docents aprenguin a integrar de forma significativa aquesta competència en la seva didàctica.

Aquest recurs està orientat a futurs i futures mestres amb l'objectiu de facilitar l'adquisició d'un coneixement bàsic i fonamental sobre el pensament computacional i la seva aplicació en contextos educatius.

1. Definició i justificació

1.1. Definició

El terme *pensament computacional* no és nou. Ja entre el 1970 i el 1980 Seymour Paper va introduir aquest terme en algunes de les seves publicacions quan va iniciar els seus treballs amb el llenguatge de programació Logo al MIT. Però no va ser fins al 2006 quan Wing va publicar un article on establia els elements clau que definien una primera aproximació al concepte de pensament computacional:

«Implica la resolució de problemes, el disseny de sistemes i la comprensió de la conducta humana, fent ús dels conceptes fonamentals de la informàtica. [...] El pensament computacional és una habilitat fonamental per a tothom, no només per als informàtics. En la lectura, escriptura i aritmètica, hauríem d'afegir el pensament computacional a la capacitat analítica de cada nen.»

Wing (2006, p. 33)

Posteriorment, altres autors i autores han enriquit aquest terme i, si bé no s'ha arribat a una única definició consensuada, la comunitat científica dona per bona la proposta elaborada el 2015 pel CSTA (Computer Science Teachers Association) i la ISTA (Society for Technology in Education) que defineix el pensament computacional com un procés de solució de problemes que inclou, entre d'altres, les característiques següents:

- Formular problemes de manera que es pugui utilitzar un ordinador o màquines per resoldre'ls.
- Organitzar i analitzar dades lògicament.
- Representar dades a través d'abstraccions com ara models i simulacions.
- Automatitzar solucions a través del pensament algorítmic (una sèrie de passos discrets i ordenats).
- Identificar, analitzar i implementar possibles solucions amb l'objectiu d'aconseguir la combinació més efectiva i eficient de passos i recursos.
- Generalitzar i transferir aquest procés de solució de problemes a una àmplia varietat de situacions.

En els darrers anys, el concepte de pensament computacional ha superat la barrera dels ordinadors per redefinir-se com un procés de pensament que no depèn necessàriament de les ciències de la computació (Bocconi *et al.*, 2016). És a dir, la competència en pensament computacional va més enllà de la tecnologia, i les habilitats que s'hi associen són aplicables en diversos contextos del dia a dia.

1. Definició i justificació

1.2. Justificació

Actualment, la majoria dels països de la Unió Europea inclouen el pensament computacional en els currículums d'educació primària i secundària obligatòria, tal com evidencia l'informe *Reviewing Computational Thinking in Compulsory Education* (Bocconi et al., 2022). En l'informe del Join Research Center elaborat per Bocconi et al. (2016) ja s'identificaven dues raons per les quals els governs tendien a incloure cada vegada més el pensament computacional en els currículums. En primer lloc, perquè els infants i joves puguin «pensar de manera diferent, expressar-se a través d'una varietat de mitjans, resoldre problemes del món real i analitzar temes quotidians des d'una perspectiva diferent». I, en segon lloc, «per impulsar el creixement econòmic, cobrir els llocs de treball TIC i preparar-se per a futurs llocs de treball».

La relació entre programació i pensament computacional és molt estreta. Això no obstant, la finalitat del pensament computacional no és aprendre a programar, sinó desenvolupar habilitats que permetin un pensament reflexiu i resolutiu de problemes, atenent necessitats socials de diferent tipus, amb l'ajuda de les eines informàtiques que tenim al nostre entorn quotidià (Ballarades; Avilés; Pérez, 2016). Per aquest motiu, la tendència dels currículums actuals és la de situar el pensament computacional en l'àmbit de la competència digital. Aquest fet es fa palès en els diferents marcs de referència de la competència digital que han publicat respectivament UNESCO (2018), JRC European Union (2017), ISTE (2021, 2017) i INTEF (2017), en els quals el pensament computacional pren protagonisme en algunes de les àrees que s'hi descriuen.

La literatura sobre el pensament computacional en edats escolars compta amb un considerable nombre de publicacions, però fins al moment no hi ha un ampli consens sobre els contextos i les didàctiques més apropiades per al seu ensenyament. Tanmateix, els autors i autores coincideixen en la seva importància en la formació d'infants i joves. Les diferents mirades convergeixen en la idea que en la societat digital el pensament computacional és de gran valor des del punt de vista educatiu, social, cultural i econòmic. Permet resoldre problemes científics i tecnològics, però també de molts altres àmbits. Fomenta la pràctica del treball col·laboratiu, la creativitat i el raonament lògic. I consolida la fluïdesa per comunicar idees de forma efectiva i creativa (Resnick i Rusk, p. 122), entre d'altres. Per això, autors com Bers et al. (2019, p. 131) defineixen el pensament computacional com una competència fonamental del segle XXI.

2. Les cinc habilitats fonamentals del pensament computacional

2.1. Introducció

Bona part dels llocs de treball que existiran en els anys vinents encara no han estat inventats. Les persones hauran de ser capaces d'adaptar-se a feines emergents i d'aplicar un conjunt d'habilitats pròpies del segle XXI, entre les quals ocupen un paper important les habilitats tecnològiques vinculades al pensament computacional (Coronel i Lima, 2020).

En aquest sentit, existeix una literatura extensa sobre les habilitats associades al pensament computacional. Bocconi *et al.* (2016) van identificar les habilitats més presents en la literatura i van confeccionar la llista següent:

- Abstracció
- Informació
- Anàlisi
- Pensament algorítmic
- Automatització
- Pensament recursiu
- Representació de dades
- Simulació
- Formulació de problemes per poder resoldre'ls amb ordinadors
- Creació de programes de computadora
- Generalització
- Avaluació i depuració

En l'àmbit educatiu, aquesta llista se sintetitza en cinc habilitats objectiu que tot estudiant de hauria d'adquirir manera gradual durant la seva escolarització obligatòria (CAS, 2015; Dorling, 2015):

- Algorítmica
- Descomposició
- Abstracció
- Generalització i patrons
- Avaluació

Aquestes habilitats esdevenen de gran valor, a més a més, perquè són aplicables en molts contextos més enllà del tecnològic. Per aquest motiu, haurien de formar part de la formació integral de tota persona per tal que pugui viure, treballar, aprendre, comunicar-se i participar en la societat de la informació (Joint Research Center, 2016).

2. Les cinc habilitats fonamentals del pensament computacional

2.2. Algoríctica

Un algorisme és una seqüència d'instruccions o un conjunt de normes per arribar a una solució. En l'elaboració d'un algorisme, cal saber definir cadascuna de les accions que cal fer i l'ordre en què s'han de fer.

En el dia a dia, apliquem algorismes constantment, per exemple en la realització d'una recepta de cuina, en donar instruccions a una persona per anar d'un punt a un altre, o quan fem la bugada.

En programació, l'algorisme és el conjunt de codis que permet arribar a una solució. És el text que descriu com s'ha de fer quelcom amb un determinat llenguatge de programació. De la mateixa manera que en qualsevol altre àmbit, l'algorisme és reutilitzable i aplicable en problemes de les mateixes característiques. Una vegada hem creat l'algorisme per respondre si un nombre és parell o senar, el podem aplicar a qualsevol nombre, de la mateixa manera que podem fer la recepta d'un pa de pessic encara que estigui en una cuina diferent. Les instruccions són les mateixes.

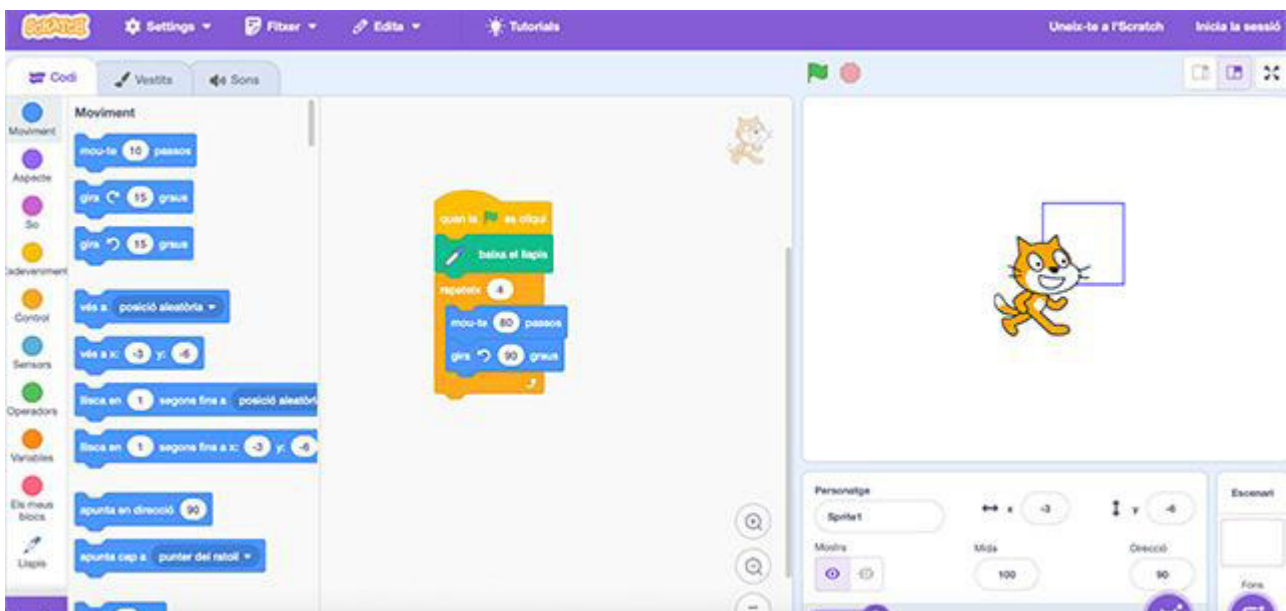


Figura 1. Algorisme
Font: [Scratch.mit.edu](https://scratch.mit.edu)

En la figura 1 es pot veure un algorisme programat amb el llenguatge Scratch del MIT.

Scratch és un entorn de programació visual desenvolupat per al MIT Media Lab que permet als estudiants crear històries interactives, jocs i animacions mitjançant l'ús de blocs de programació en lloc de codi de programació tradicional. Això fa que aprenguin els conceptes bàsics de la programació, com ara seqüències d'instruccions, bucles i condicionals, i apliquin les habilitats bàsiques del pensament computacional, d'una manera visual, intuïtiva i divertida.

En aquest cas concret de la figura 1, el programa indica bloc a bloc els passos necessaris per dibuixar un quadrat a la pantalla. El primer bloc executa el programa. El segon baixa el llapis –indispensable per poder dibuixar. El tercer bloc correspon a una repetició: un quadrat té quatre costats i quatre angles iguals, i, per tant, dibuixarem quatre vegades el mateix. Els blocs quart i cinquè responen al dibuix d'un costat i a un gir de noranta graus respectivament. Aquest algorisme amb el llenguatge de programació Scratch serveix per dibuixar qualsevol quadrat. Podem canviar-ne la mida, però, mentre sigui un quadrat, sempre serà un algorisme vàlid.

2. Les cinc habilitats fonamentals del pensament computacional

2.3. Descomposició

En la resolució de problemes complexos, una habilitat fonamental és saber dividir aquest problema en problemes més senzills, la resolució dels quals condueix a la solució.

La descomposició és una habilitat aplicada en molts àmbits de la vida, com per exemple en l'organització d'un viatge familiar (definició del pressupost, compra dels bitllets d'avió, reserva de les habitacions d'hotel, programació de les jornades, etc.). Sovint a les classes de matemàtiques s'aplica de manera natural quan un problema requereix diverses operacions per trobar dades, les quals permeten fer altres operacions que donaran lloc a la resposta final.

La descomposició és una habilitat clau per fer front a la programació de qualsevol projecte. Un equip de programadors no pot abordar la programació d'un videojoc com una sola tasca, per exemple. Requereixen una planificació de treball que divideixi el videojoc en tasques més petites i assumibles, la consecució de les quals permet anar completant el projecte d'una manera raonable. En aquest mateix sentit, l'alumnat de primària necessita aplicar aquesta habilitat per descompondre les tasques complexes plantejades en una sèrie de passos i anar-se apropant a la solució final d'una manera més senzilla i encoratjadora.

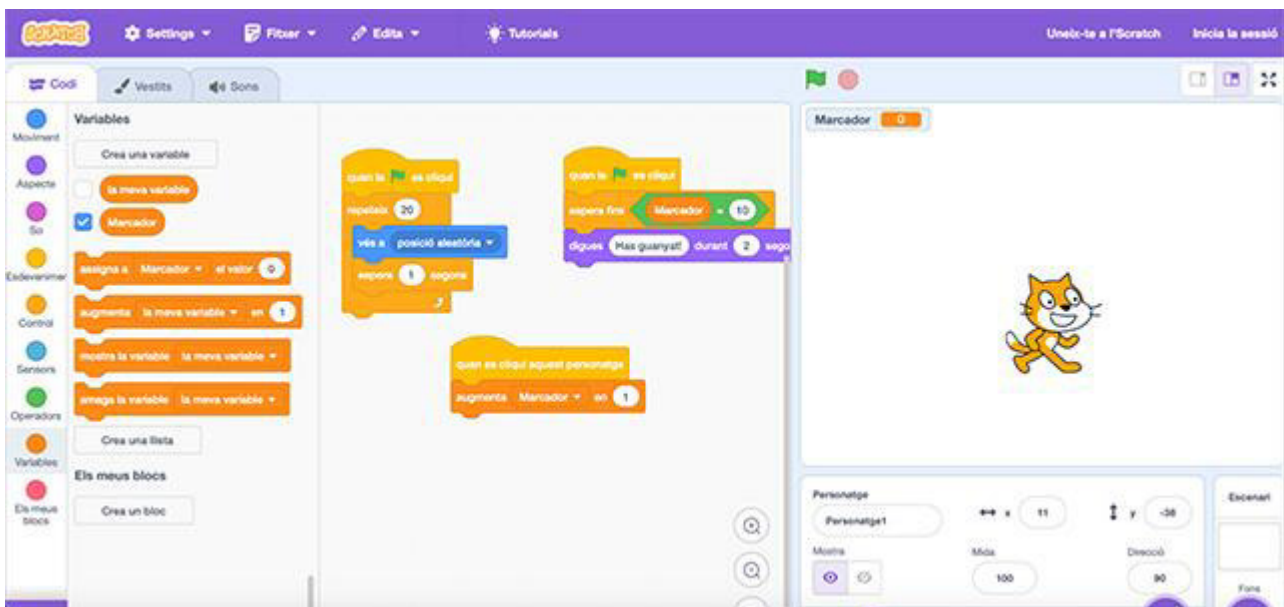


Figura 2. Descomposició

Font: [Scratch.mit.edu](https://scratch.mit.edu)

En la figura 2 es presenta un joc elaborat amb Scratch. L'objectiu del joc consisteix a arribar als deu punts clicant el personatge que es va movent aleatòriament per la pantalla a una velocitat d'una posició per segon. Aquesta tasca la podem fragmentar en tres reptes més senzills:

1. Elaborar un codi perquè el personatge es mogui aleatòriament per la pantalla cada segon.
2. Crear un marcador perquè augmenti el seu valor quan es cliqui el personatge.
3. Establir una condició perquè el programa estigui pendent de quan el marcador arribi a deu punts. En aquest moment, el personatge ha de dir «Has guanyat!».

D'aquesta manera, la realització del joc se simplifica resolent tres reptes que per separat són més assumibles.

2. Les cinc habilitats fonamentals del pensament computacional

2.4. Abstracció

L'abstracció és un habilitat clau del pensament computacional. Consisteix a identificar el que és realment rellevant i descartar tot allò que no és estrictament necessari. D'aquesta manera, la informació important esdevé més comprensible i, alhora, esdevé més simple centrar-se en el que és essencial.

Per exemple, en les estacions de metro sovint podem trobar una representació de les estacions damunt d'una línia recta. En aquesta representació hi apareixen de manera ordenada les estacions de la línia, s'hi assenyalen l'estació on ens trobem i s'indiquen les correspondències amb altres línies. S'ha obviat el recorregut real que fa el metro: totes les estacions estan disposades en línia recta. També s'han obviat les distàncies entre estacions, ja que en la realitat no són totes equidistants. Però aquesta abstracció permet a les persones identificar on es troben, quantes parades han de fer per arribar a la seva destinació o a quina estació han de baixar per canviar de línia.

Un altre exemple el trobem en el menú diari dels restaurants. En un sol full de paper es llisten unes opcions de primer plat, unes opcions de segon plat i les postres, amb els corresponents preus. No s'especifiquen tots els ingredients que s'utilitzen en cada plat ni el procés que se segueix per cuinar-los. Però aquesta simplificació se centra en l'essencial, i així, de forma ràpida i senzilla, els clients i les clientes poden triar el dinar per un preu tancat.

A l'hora de programar, l'abstracció permet centrar-se en el nucli del programa. Permet identificar els blocs de codi a partir del qual es crearà el programa sencer. En la figura 3 es mostra un programa per dibuixar un pentàgon regular d'un color i una mida determinats. A partir d'aquest exemple es demana a l'alumnat que creï un programa per dibuixar un hexàgon.

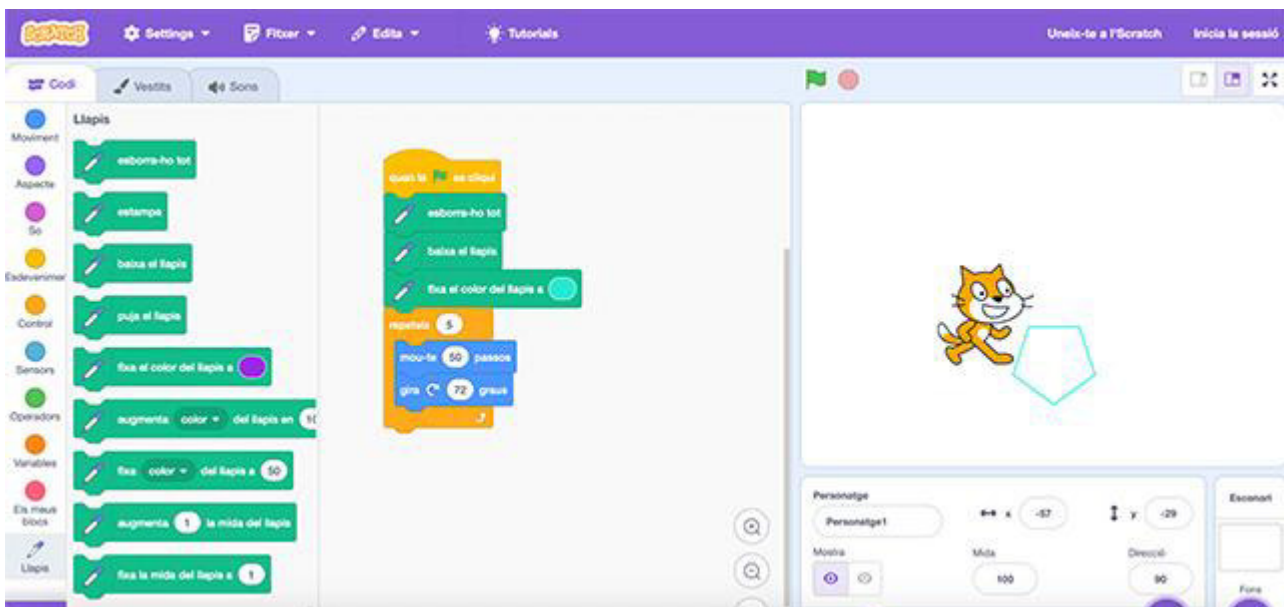


Figura 3. Abstracció
Font: [Scratch.mit.edu](https://scratch.mit.edu)

Analitzant el programa, podem observar que qualsevol polígon es podrà dibuixar fent una repetició de tantes vegades com costats tingui el polígon i fent un gir en graus equivalent a dividir una volta sencera, 360 graus, pel nombre de costats, ja que el dibuix comença i acaba en el mateix punt. En la taula següent es mostren tres exemples:

Taula 1. Aplicació de l'abstracció

	Nombre de costats	Repeticions	Gir en graus (360/costats)
Pentàgon	5	5	$360/5 = 72$
Hexàgon	6	6	$360/6 = 60$

Font: elaboració pròpia.

En aquest cas, doncs, l'elaboració del programa es redueix a identificar el nombre de costats del polígon i a dividir 360 graus per aquest nombre de costats. Aquest és el cor d'aquesta programació i el codi es programarà a partir d'aquest concepte central. Caldrà aplicar altres conceptes computacionals i habilitats per completar-lo, però el nucli és aquesta abstracció.

2. Les cinc habilitats fonamentals del pensament computacional

2.5. Generalització i patrons

La identificació de patrons permet fer prediccions, establir normes i resoldre problemes que comparteixen una mateixa analogia.

Troblem patrons en moltes situacions del dia a dia. A la carretera, els serveis de trànsit tenen molt clar un patró que es repeteix gairebé cada setmana: divendres molts vehicles surten de les grans ciutats i diumenge hi tornen. Aquest patró permet anticipar mesures per millorar el trànsit: obrir carrils alternatius, avisar als usuaris que facin una sortida o tornada esglaonada, activar més serveis d'emergència per si hi ha algun accident, etc.

Al carrer trobem un altre exemple de patrons, en aquest cas per fer un programa eficient. Els semàfors segueixen una successió d'il·luminació (verd, taronja, vermell) en uns intervals de temps determinats. La programació es basa en una repetició d'aquesta sèrie de colors al llarg del dia. N'hi ha prou amb programar-la una vegada i fer que el programa es repeteixi permanentment. És a dir, la identificació del patró redueix la programació a la seva escriptura una sola vegada.

En programació, els patrons són bàsics i fonamentals per elaborar programes més eficients i reutilitzar programes existents per a contextos de les mateixes característiques. Quan identifiquem patrons i aquests patrons ens permeten establir una resposta comuna per a problemes de la mateixa analogia, diem que hem arribat a una generalització.

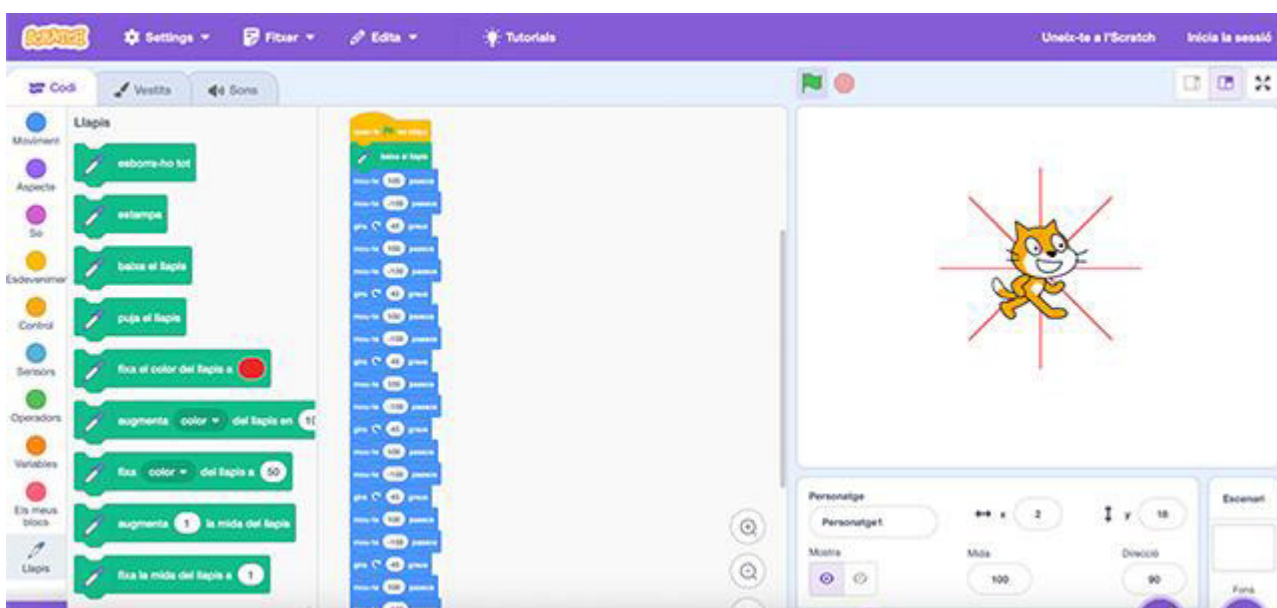


Figura 4. Cerca de patrons
Font: [Scratch.mit.edu](https://scratch.mit.edu)

En aquest programa es dibuixa a la pantalla una figura en forma d'estrella de vuit puntes. Una lectura atenta del programa permet observar que hi ha tres blocs de codi que es repeteixen successivament: mou-te endavant (100), mou-te endavant (-100) i gira (45°). Aquest conjunt de blocs formen un patró i es repeteixen vuit vegades. Això vol dir que aquest programa es pot escriure de forma més eficient, introduint un bloc de repetició (bucle) de la manera següent:

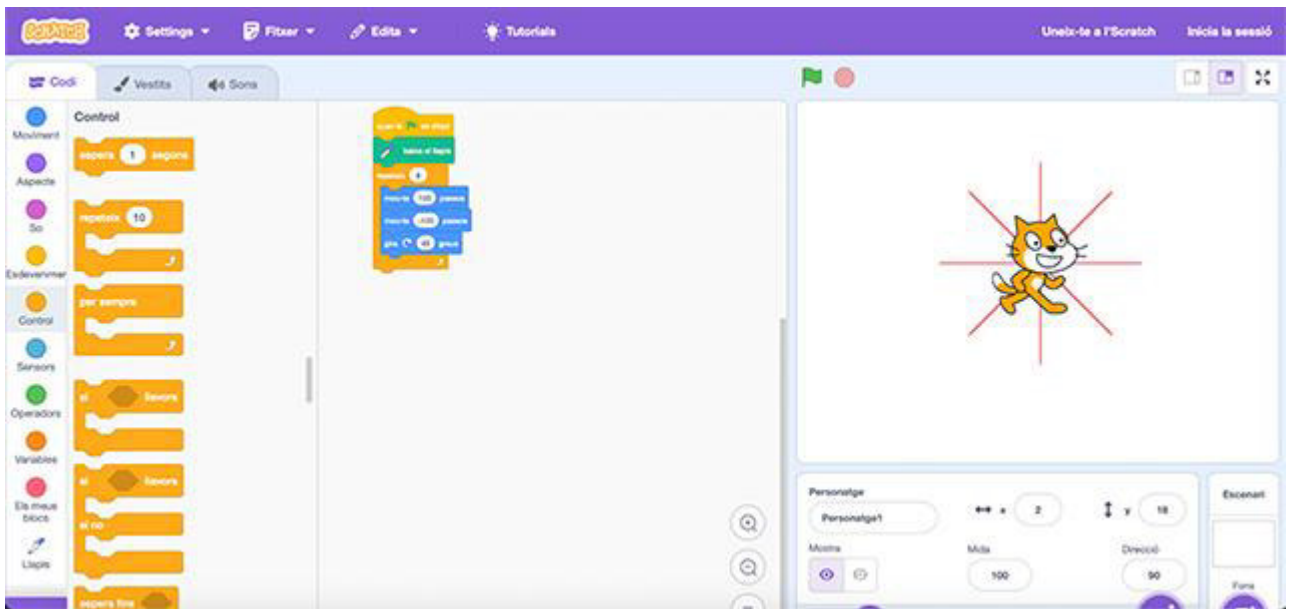


Figura 5. Aplicació de patrons
Font: [Scratch.mit.edu](https://scratch.mit.edu)

2. Les cinc habilitats fonamentals del pensament computacional

2.6. Avaluació

El fet d'avaluar l'apliquem constantment en moltes activitats de la nostra vida. De la mateixa manera que és important una avaluació externa que validi la nostra feina, ho és també la capacitat com a persones de valorar nosaltres mateixes i objectivament allò que hem fet. No es tracta d'una avaluació quantitativa sinó qualitativa. És recordar l'objectiu amb el qual iniciàvem una tasca i valorar si el resultat és bo, si requereix millores o si no és correcte.

Per exemple, si hem pintat una habitació de casa nostra, valorem el resultat final. Estan totes les parets ben pintades? El color és uniforme? Les cantonades i racons han quedat bé? Aquesta anàlisi ens ha de permetre donar la feina per bona i, per tant, per acabada, o bé portar a terme accions complementàries per obtenir el resultat esperat.

En altres situacions, pot donar-se el cas que, si bé el resultat és correcte, no és la millor opció.

Posem per cas que una escola decideix obrir una nova porta d'accés més segura per al seu alumnat. El resultat final és molt bo en molts sentits: la porta és més gran, la vorera del davant és àmplia i separada dels vehicles per una barana, i per a les famílies és més fàcil localitzar els seus fills i filles. Però la nova porta respecte al carrer té un petit desnivell, que s'ha salvat amb una escala de dos graons. L'avaluació d'aquesta solució permet observar que l'alumnat amb cadira de rodes i els familiars amb cotxet de nadons tindran un impediment important per accedir a l'escola.

En programació, l'avaluació és una habilitat imprescindible per assolir els millors resultats i la màxima eficiència. Si concebem la programació com un diagrama de flux, l'avaluació seria sempre el pas previ a la finalització. Si l'avaluació és bona, hem acabat. Si l'avaluació no és prou bona, tornem a l'objectiu inicial i revisem la feina per millorar la solució.

En la figura 6 es mostra un repte d'Scratch en el qual es demana dibuixar una circumferència de color blau. Tal com es pot veure a l'àrea dreta, el resultat és correcte. Això no obstant, aplicant l'habilitat d'avaluació, en el codi s'identifica un error.

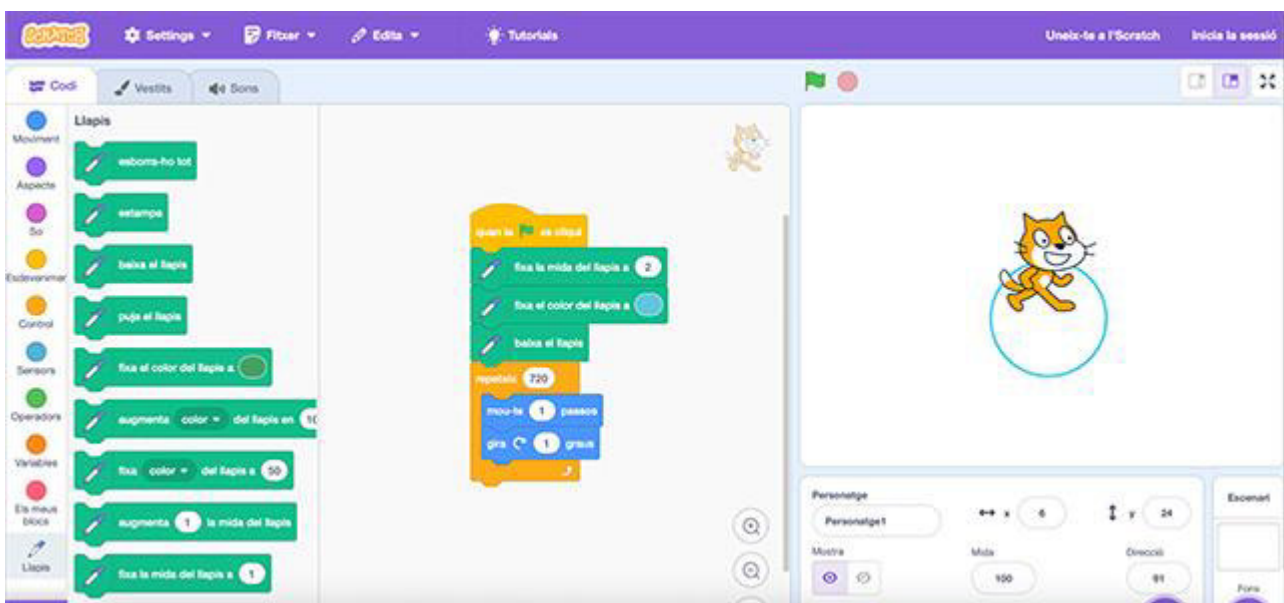


Figura 6. Avaluació
Font: [Scratch.mit.edu](https://scratch.mit.edu)

El dibuix de la circumferència es pot traçar amb un moviment repetitiu d'un píxel i un grau 360 vegades, que corresponen als 360 graus d'una volta sencera. En aquest cas, però, la repetició és de 720 vegades, la qual cosa implica que es tracen dues voltes dibuixant dues circumferències idèntiques l'una sobre l'altra. Si traslladem aquesta programació al món industrial, podria tractar-se d'un programa que cus una circumferència en una samarreta. Això vol dir que en la producció de cada samarreta s'estaria utilitzant el doble de fil per unitat, i això, multiplicat per milers de samarretes, seria una despesa considerable. Per aquest motiu,

l'avaluació és una habilitat molt important en el pensament computacional. Cal saber valorar la feina feta i determinar objectivament si s'ha trobat la millor solució per a l'objectiu plantejat.

2. Les cinc habilitats fonamentals del pensament computacional

2.7. Resum d'idees clau

Les habilitats de pensament computacional cal concebre-les també com un tot, en el qual les habilitats estan interrelacionades i s'apliquen sovint de forma integrada. En aquestes relacions, l'avaluació és un punt de convergència amb totes les altres habilitats per establir sovint la necessitat d'aplicar o repetir alguna habilitat amb l'objectiu d'assolir un millor resultat.

En resum, les idees clau que defineixen les habilitats de pensament computacional són les següents:

Habilitat	Descripció
Algorítmica	Desenvolupar un conjunt d'instruccions pas a pas per resoldre un problema o fer una tasca de manera eficient i efectiva.
Descomposició	Dividir un problema complex en parts més petites i manejables per comprendre millor el problema i abordar-lo de manera més efectiva.
Abstracció	Identificar els aspectes essencials d'un problema i ignorar els detalls no rellevants. Utilitzar models i representacions per simplificar i comprendre el problema.
Generalització i patrons	Identificar tendències o seqüències regulars en dades o problemes i utilitzar-les per predir o resoldre situacions noves.
Avaluació	Revisar el resultat final d'una tasca i valorar objectivament si s'ha complert d'acord amb el seu objectiu inicial. En cas negatiu, identificar quines millores caldria aplicar per assolir un resultat satisfactori.

3. Connexions entre el pensament computacional i àmbits curriculars de primària

Els exemples explicats juntament amb les habilitats del pensament computacional ens han permès copsar que aquestes habilitats es poden aplicar en molts àmbits de la vida, i, per aquest motiu, el seu valor traspasa la frontera dels entorns de programació per esdevenir un conjunt d'habilitats clau per a la ciutadania. Si bé existeixen diversos entorns de tecnologia digital (com Scratch o Code) que permeten fer un treball sistemàtic del pensament computacional, hi ha altres àmbits curriculars que permeten treballar el pensament computacional a través d'activitats concretes.

En aquest sentit, a continuació es presenten exemples d'activitats basades en les habilitats de pensament computacional emmarcades en diferents àmbits curriculars.

Algorística	Descomposició
<ul style="list-style-type: none"> • Escriure els passos per fer una multiplicació italiana. • Elaborar un diagrama per explicar la fotosíntesi. • Explicar en vídeo els passos per fer un model de papiroflèxia. • Elaborar un tutorial per editar una imatge. • Redactar les instruccions d'un joc. • Crear un guió il·lustrat d'un curt. • Escriure una recepta de cuina. • Redactar els passos per resoldre un tipus de problema matemàtic. • Dibuixar un recorregut en un mapa. • Inventar els moviments per a una coreografia. • Elaborar una línia del temps. • Redactar els passos per a la realització d'un experiment. • Seqüenciar en fotografies la construcció d'una maqueta. • Fer un arbre genealògic. • Crear una sèrie d'exercicis d'escalfament per a una classe d'educació física. 	<ul style="list-style-type: none"> • Planificar un text amb les idees clau d'acord amb una estructura del tipus inici, nus i desenllaç. • Dividir un problema matemàtic complex en parts més senzilles. • Identificar les comarques de Catalunya: primer les de costa, després les de muntanya i finalment les d'interior. • Organitzar un projecte de treball en una successió de tasques. • Llistar el necessari per organitzar una festa. • Identificar els instruments presents en una cançó. • Identificar les dades necessàries per calcular una àrea. • Fraccionar el recorregut d'un robot en trams. • Llistar els ingredients per elaborar una recepta de cuina. • Elaborar un checklist per completar una tasca complexa. • Organitzar la visita a una ciutat a partir dels seus punts d'interès. • Llistar el contingut per a una motxilla de colònies. • Dividir una cançó en parts per aprendre-la. • Explicar les posicions dels jugadors d'un equip de bàsquet. • Explicar les etapes que segueix un producte quan el comprem per internet.
Abstracció	Generalització i patrons
<ul style="list-style-type: none"> • Resumir el cicle de l'aigua en cinc idees. • Llistar els elements clau d'un tema treballat. • Elaborar un mapa conceptual sobre els tipus de quadrilàters. • Descriure una imatge. • Explicar el més destacat del cap de setmana. • Presentar una idea per a un projecte en menys d'un minut. • Explicar qui va ser Thomas Alba Edison. • Identificar les dades útils per a la resolució d'un problema matemàtic. • Fer un croquis sobre el mapa d'un país. • Elaborar un decàleg sobre la internet segura. • Fer un esbós sobre una maqueta. • Explicar l'argument d'una pel·lícula. • Convertir un text informatiu en una infografia. • Crear un còmic per explicar la llegenda de sant Jordi. • Fer un resum sobre un partit de voleibol escolar. 	<ul style="list-style-type: none"> • Deducir una norma ortogràfica a partir de textos on s'aplica. • Identificar patrons en sèries alfanumèriques o d'imatges. • Identificar patrons musicals dins d'una cançó. • Fer textures en dibuixos a partir de patrons. • Descobrir l'element que no compleix el patró dins d'una sèrie. • Jugar al SET. • Classificar objectes d'acord qualitats específiques: color, forma, mida, etc. • Aplicar una estratègia coneguda en un problema matemàtic. • Detectar la freqüència d'un tren a partir dels seus horaris. • Aplicar un bucle per a blocs de programació iguals. • Autoavaluar-se a partir d'un treball de referència. • Classificar animals segons el seu grup de vertebrats. • Preveure fets d'acord amb patrons observats. • Completar figures de tangram. • Treballar rutines per consolidar hàbits.

Avaluació
<ul style="list-style-type: none"> • Utilitzar un <i>checklist</i> per revisar la feina un cop es finalitza. • Compartir els objectius a l'inici de l'activitat i revisar-los en acabar. • Definir uns criteris que permetran valorar si una tasca s'ha complert d'acord el seu objectiu inicial. • Avaluar el treball d'una altra persona a partir d'una rúbrica. • Elaborar una crítica sobre un llibre. • Escollir la millor opció d'entre diverses argumentant la resposta. • Elaborar un portafolis durant la realització d'un tema o un projecte. • Optimitzar el codi d'una programació perquè sigui més eficient. • Analitzar dades per prendre una decisió. • Llistar els punts forts i febles sobre un tema de debat. • Identificar errades en la resolució d'uns problemes matemàtics. • Descriure la metacognició associada a un procés d'aprenentatge. • Autoavaluar la feina individual en el marc d'un treball en equip. • Enregistrar-se en vídeo i valorar els aspectes destacats i els aspectes a millorar. • Innovar un aspecte de l'escola per millorar-lo.

En els enllaços següents desenvolupem els exemples de les cinc habilitats fonamentals del pensament computacional:

- [Algorística](#)
- [Descomposició](#)
- [Abstracció](#)
- [Generalització i patrons](#)
- [Avaluació](#)

Conclusió

En el decurs d'aquest material s'ha fet palès que el pensament computacional és una competència essencial al segle XXI. Inclou un conjunt d'habilitats clau per a la resolució de problemes, per a l'ús eficient de llenguatges de programació i també per a molts altres àmbits de la vida. Per aquest motiu, és cabdal que tot docent integri la seva pràctica en les seves programacions didàctiques. En aquest sentit, Bocconi *et al.* (2016) suggereixen que el pensament computacional es pot integrar de diverses maneres en el currículum: com a tema transversal, com a assignatura separada, dins d'altres assignatures com les matemàtiques o la tecnologia, o, fins i tot, de forma separada de l'àmbit digital.

El que és important és definir com a centre educatiu una seqüenciació del treball del pensament computacional al llarg dels cursos, establint una coherència vertical i assegurant una bona competència de l'alumnat en acabar l'etapa d'educació primària. És per això que diversos autors subratllen la necessitat de formar els equips de mestres perquè adquireixin una bona experiència didàctica en aquesta matèria, no només per esdevenir bons guies en el corresponent aprenentatge dels alumnes, sinó també per integrar i treballar el pensament computacional en el dia a dia i des de diferents perspectives i situacions.

Bibliografia

Bocconi, S., Chiocciariello, A., Dettori et al. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*, EUR 28295 EN. <https://data.europa.eu/doi/10.2791/792158>

CAS, Computing at School (2015). *CAS Barefoot* [Pàgina web]. <http://barefootcas.org.uk/>

Dorling, M. (2015). *CAS Computing Progression Pathways KS1 (Y1) to KS3 (Y9) by topic* [Document en línia]. <http://community.computingatschool.org.uk/resources/1692>

García-Peñalvo, F. i Rojas-López, A. (2020). Evaluación de habilidades del pensamiento computacional para predecir el aprendizaje y retención de estudiantes en la asignatura de programación de computadoras en educación superior. *RED. Revista de Educación a Distancia*, 63(20), art. 4. DOI: <http://dx.doi.org/10.6018/red.409991>

Loureiro, A.C. et al. (2022). Computational thinking in teacher digital competence frameworks. *Revista Prisma Social*, (38), p. 77-93. <https://revistaprismasocial.es/article/view/4783>

Polanco, N., Ferrer, S. i Fernández, M. (2021). Aproximación a una definición de pensamiento computacional. *RIED. Revista Iberoamericana de Educación a Distancia*, 24(1), p. 55-76. DOI: <http://dx.doi.org/10.5944/ried.24.1.27419>

Soria, E. i Rivero, C. (2019). Pensamiento computacional: una nueva exigencia para la educación del siglo XXI. *Revista Espaço Pedagógico*, [S. l.], 2(26) p. 323-337. <https://seer.upf.br/index.php/rep/article/view/8702>