

UNITAT DIDÀCTICA #2

LA LÒGICA DE LA PROGRAMACIÓ

GUIA PER AL DOCENT

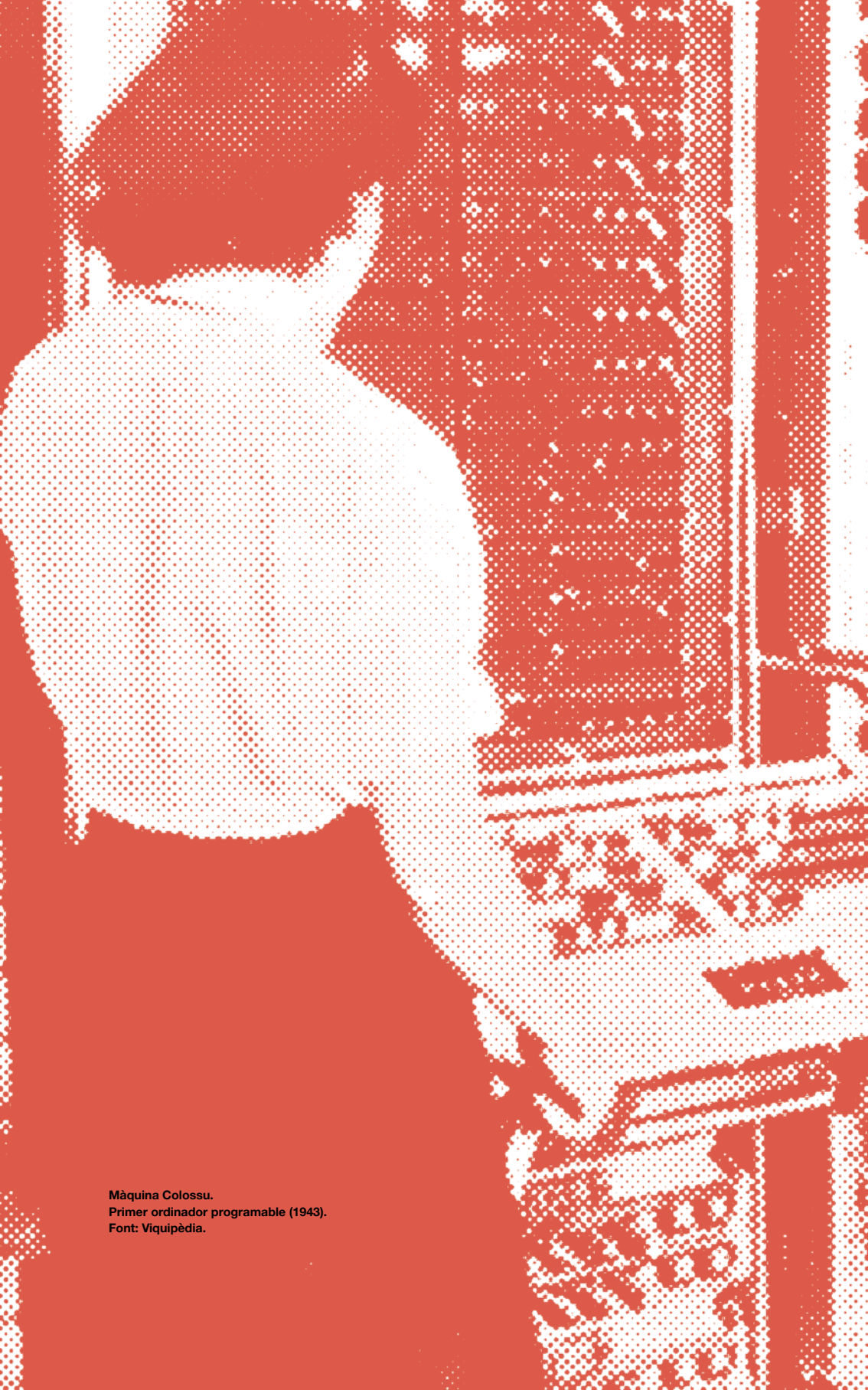
The logo consists of the letters 'U' and 'i' in a bold, white, sans-serif font. The 'U' is significantly larger than the 'i'. The 'i' has a small square dot above it. The entire logo is centered within a white square frame.

Ui

UNIVERS
INTERNET

ÍNDEx

Presentació de la unitat didàctica	6
<ul style="list-style-type: none">• Objectius específics de la unitat• Objectius generals de l'etapa• Treball de competències bàsiques• Connexions amb altres àrees• Material necessari• Preparació del professorat	
Marc teòric	12
<ul style="list-style-type: none">• Introducció• Què vol dir programar?<ul style="list-style-type: none">• Llenguatge• Codi• Pensament computacional• Conceptes clau	
Dinàmica de l'activitat	21
<ul style="list-style-type: none">• Introducció: el laberint• Llenguatge i codi: comunicar-nos amb els ordinadors<ul style="list-style-type: none">• Llenguatge<ul style="list-style-type: none">• Malentesos quotidians• Dibuixa les instruccions• Codi<ul style="list-style-type: none">• Codis quotidians• Codi binari• Codi laberint• Algoritmes: sortir del laberint• Tancament de l'activitat	
Proposta d'avaluació de l'activitat	36
Orientacions metodològiques	37
Material complementari	38



Màquina Colossu.
Primer ordinador programable (1943).
Font: Viquipèdia.

LA LÒGICA DE LA PROGRAMACIÓ

Aquesta unitat sorgeix d'un taller desenvolupat al **CCCB** per **Irene Lapuente**, de **La Mandarina de Newton**, dins del marc del projecte **Univers Internet**.

Temporització

3 hores

Nivell

ESO

Àrees

Tecnologia, informàtica i matemàtiques

“

El programador, com el poeta, només funciona una mica enretirat del pensament aplicat a objectes purs. Construeix els seus castells en l'aire, des de l'aire, creant a partir de l'esforç de la imaginació. Pocs mitjans de creació són tan flexibles, tan fàcils de polir i modelar, i capaços de copsar grans estructures conceptuals.

FREDERICK BROOKS

”

PRESENTACIÓ DE LA UNITAT DIDÀCTICA

Aquesta unitat presenta un recorregut que va des de l'anàlisi de diferents codis i llenguatges fins al pensament computacional, per tal d'entendre com es comuniquen els ordinadors.

OBJECTIUS ESPECÍFICS DE LA UNITAT

En acabar la unitat, els alumnes haurien de ser capaços de:

- Entendre la diferència entre llenguatges naturals i formals.
- Analitzar diferents codis i crear-ne de nous.
- Entendre el sistema binari i explicar per què s'utilitza en la computació.
- Conèixer les característiques principals del pensament computacional.
- Reconèixer patrons i abstroure'n variables.
- Interpretar i formular algoritmes senzills.
- Aplicar la lògica del pensament computacional a diferents situacions de la vida quotidiana i acadèmica.

OBJECTIUS GENERALS DE L'ETAPA

OBJECTIU	COM ES TREBALLA?
Desenvolupar i consolidar hàbits d'esforç, d'estudi, de treball individual i cooperatiu i de disciplina com a base indispensable per a un aprenentatge eficaç i per aconseguir un desenvolupament personal equilibrat.	La seqüència d'activitats i la seva dinàmica estan pensades perquè siguin els mateixos alumnes, conduïts pel professor, els que adquireixin el coneixement a través de formular-se preguntes. El pensament computacional ajuda a treballar de manera sistemàtica.
Desenvolupar l'esperit emprenedor i la confiança en si mateix, la participació, el sentit crític, la iniciativa personal i la capacitat per aprendre a aprendre, planificar, prendre decisions i assumir responsabilitats.	El pensament computacional dóna eines als alumnes a fi que siguin capaços d'analitzar per si sols els problemes a resoldre i crear un pla per solucionar-los.
Adquirir coneixements bàsics que capacitin per a l'exercici d'activitats professionals i alhora facilitin el pas del món educatiu al món laboral.	Les professions relacionades amb la informàtica i la programació tenen una gran demanda, però encara no es treballen prou a les etapes escolars obligatòries. Aquesta activitat busca despertar l'interès dels alumnes per les ciències de la computació.

TREBALL DE COMPETÈNCIES BÀSIQUES

COMPETÈNCIA	COM ES TREBALLA?
Competència comunicativa lingüística i audiovisual.	En aquesta unitat s'expliquen conceptes fonamentals de la comunicació: què és un llenguatge, la diferència entre llenguatge natural i formal o què són els codis. També es treballen qüestions pràctiques com la precisió del llenguatge.
Competència matemàtica.	Treballar amb el sistema binari obliga a plantejar-se qüestions bàsiques de les matemàtiques, com ara quina és la lògica del sistema decimal. A més, amb els exercicis proposats es treballa el càlcul mental.
Competència d'aprendre a aprendre.	Els alumnes descobriran les respostes per ells sols, no serà el professor qui transmeti el coneixement. Aquest sistema dóna eines a fi que els alumnes puguin aprendre de manera autònoma.

TREBALL DE COMPETÈNCIES BÀSIQUES

COMPETÈNCIA	COM ES TREBALLA?
Competència d'autonomia i iniciativa personal.	El pensament computacional dóna eines als alumnes per resoldre problemes per si sols.
Competència en el coneixement i la interacció amb el món físic.	Aquesta activitat té una forta vinculació amb el dia a dia dels alumnes, ja que analitza els principis de la informàtica, que és omnipresent en les nostres vides.

CONNEXIONS AMB ALTRES ÀREES

Aquesta unitat didàctica treballa continguts de diverses matèries, principalment de l'àrea de tecnologia, informàtica i matemàtiques.

MATERIAL NECESSARI



- **Prezi** amb la guia de la unitat
- Cinta aïllant o guixos grans per dibuixar a terra o pintura de terra per dibuixar els laberints
- Fitxes impreses:
 1. Laberints a l'aula
 2. Laberints al pati
 3. Malentesos quotidians
 4. Dibuixa les instruccions
 5. Codi quotidians
 6. Codi binari
 7. Codi laberint
 8. Algoritme de Trémaux
- Llapis
- Pedres o cigrons per marcar el pas pel laberint

PREPARACIÓ DEL PROFESSORAT

1. Llegir la guia i consultar el material complementari recomanat.
2. Escollir els exercicis que es vulguin realitzar.
3. Imprimir o fotocopiar les fitxes necessàries.

* *Escriu un email a seducatiu@cccb.org i t'enviarem els fitxers.*

“

En el procés d'aprendre a codificar, les persones aprenen moltes altres coses. No només estan aprenent codi, s'estan codificant per aprendre. Aquestes habilitats són útils no només per als científics de la computació, sinó per a tothom, independentment de l'edat, els antecedents, els interessos o l'ocupació.

MITCHEL RESNICK

”

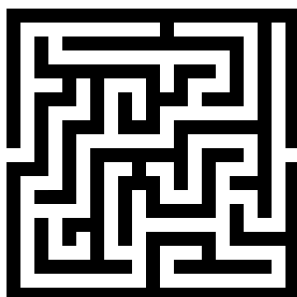
MARC TEÒRIC

INTRODUCCIÓ

El punt d'origen d'una bona programació és la definició de quin és el problema a resoldre, i a partir d'aquí descompondre el procés per arribar-hi i construir tots els mòduls que cal per fer-ho possible. Com podem aplicar aquesta lògica a la vida quotidiana? I en el plantejament d'un treball acadèmic per obtenir més bons resultats? Com s'aplica a la computació?

En aquesta unitat didàctica treballarem alguns conceptes propis del pensament computacional (*Computational Thinking*). Farem un recorregut que ens portarà des dels codis i els llenguatges naturals fins a les habilitats pròpies de la computació que es desenvolupen majoritàriament durant la fase de la resolució de problemes (*problem solving*), com ara: descomposició, reconeixement de patrons, abstracció i algorismes. També treballarem la fase de la definició de problemes (*problem discovery*).

Per tal de treballar aquests conceptes a l'aula, es proposen una sèrie d'activitats relacionades amb la resolució de laberints.



QUÈ VOL DIR PROGRAMAR?

Programar és dissenyar, codificar, depurar i mantenir el codi font de programes computacionals. Els programes computacionals són conjunts d'instruccions que, un cop s'executen, fan que una màquina realitzi una sèrie d'accions.

Per poder programar necessitem dos elements fonamentals:

- Un llenguatge i el seu codi.
- Una lògica: unes normes per transmetre la informació i que la màquina ho entengui.

LLENGUATGE

Un llenguatge és un conjunt de paraules o seqüències de símbols que pertanyen a un conjunt finit. Existeixen dos tipus bàsics de llenguatges: els naturals i els formals. Els primers tenen un origen i desenvolupament natural, és a dir, primer sorgeix el llenguatge i posteriorment se n'estableix la teoria i gramàtica. En canvi, els llenguatges formals es desenvolupen a partir d'una teoria creada *a priori* i que en configurarà la base. El primer que cal tenir en compte quan tractem el tema de la programació és que aquesta disciplina utilitza un llenguatge formal, mentre que, per comunicar-nos entre els humans, utilitzem llenguatges naturals.

LLENGUATGE NATURAL

El llenguatge natural és la llengua parlada pels humans per realitzar un acte de comunicació. Pot ser sonor o de símbols. Està dotat d'una sintaxi (regles) i no depèn de la forma, sinó de la semàntica o el significat, ja sigui específic o dins d'un context. Per tant, per considerar que una frase és correcta, el seu significat global ha de ser coherent, és a dir, ha de ser consistent amb l'experiència pràctica humana.

La frase «He llegit una patata» seria incorrecta perquè no té cap sentit, ja que les patates no es llegeixen.

Exemples de llenguatge natural són el català, el castellà, l'anglès o el rus.

LLENGUATGE FORMAL

El llenguatge formal és un llenguatge on els símbols primitius (o l'alfabet) i les seves regles (gramàtica formal o sintaxi) estan formalment especificats.

Per exemple, un llenguatge formal es podria crear a partir de l'alfabet format pel conjunt dels nombres primers de l'1 al 7, per tant: 1, 2, 3, 5, 7, i una gramàtica es podria definir com les fórmules ben formulades de qualsevol producte d'aquests nombres (1x2, 2x3, 2x5, 3x5, 3x7, etc.). En aquest cas, per tant, el llenguatge formal inclouria tots els productes possibles entre aquests nombres.

La frase «He llegit una patata» seria correcta perquè la forma és correcta, és a dir, no hi ha cap error sintàctic i tot concorda.

Són exemples de llenguatge formal el matemàtic, el de la lògica formal o els llenguatges de programació com Javascript o C.

CODI

En la comunicació, un codi és una norma per convertir un fragment d'informació (per exemple, una lletra, paraula o frase) en una altra forma de representació, no necessàriament del mateix tipus. El procés de codificació permet transformar la informació origen en unes dades codificades, que s'emmagatzemen o s'envien a un receptor. El procés invers és el de descodificació, on es converteixen aquests codis en informació interpretable per al destinatari.

CODI BINARI

Atès que els ordinadors treballen internament amb dos nivells de voltatge (encès i apagat), el sistema que utilitzen per codificar la informació està basat en només dos símbols, el 0 i l'1, que segueixen un sistema de numeració de base 2 (binari), mentre que el que utilitzem habitualment és de base 10 (decimal). Tant el sistema decimal com el binari són llenguatges formals.

El sistema binari és un sistema posicional. Per obtenir la seva representació equivalent en el sistema decimal, cal multiplicar el 0 o l'1 que hi hagi en cada posició per 2 elevat a l'exponent que correspongui: començant de dreta a esquerra, cada posició té un exponent que supera una unitat a l'anterior, sent 0 el primer de tots. Finalment, cal sumar tots els resultats i obtenim el valor representat.

Exemple:

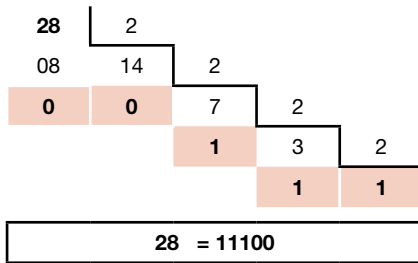
$$1100 = 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 = 0 + 0 + 4 + 8 = 12$$

$$1001 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 1 + 0 + 0 + 8 = 9$$

UNITAT DIDÀCTICA # 2 – LA LòGICA DE LA PROGRAMACIÓ

BINARI	$2^4=16$		$2^3=8$		$2^2=4$		$2^1=2$		$2^0=1$		DECIMAL
01100	0	+	1	+	1	+	0	+	0	=	12
01001	0	+	1	+	0	+	0	+	1	=	9
00111	0	+	0	+	1	+	1	+	1	=	7
10011	1	+	0	+	0	+	1	+	1	=	19
11100	1	+	1	+	1	+	0	+	0	=	28

En el procés invers, per obtenir la representació binària d'un valor, cal dividir aquest valor entre 2 successivament fins que el dividend ja no es pugui dividir. Per obtenir la representació en binari, s'agafa l'últim quocient i, a continuació, totes les restes de les divisions en sentit invers.



Per poder representar textos amb el sistema binari, s'han estandarditzat diferents codis. Els més habituals són l'ASCII (que assigna valors numèrics del 0 al 127 a les lletres, xifres i signes de puntuació). Cada valor numèric es representa amb un número binari de set xifres. Hi ha codis ASCII extensos que utilitzen números binaris de vuit xifres i així poden representar valors numèrics del 0 al 256, que permeten afegir més símbols, com ara accents i la ç, que no es troben representats en l'ASCII, ja que va ser creat per a la llengua anglesa.

En el codi ASCII, els caràcters imprimibles són tots els que corresponen al número 32 i superiors. Aquesta és la taula de conversió:

UNITAT DIDÀCTICA # 2 – LA LòGICA DE LA PROGRAMACIÓ

BINARI	DECIMAL	SÍMBOL	BINARI	DECIMAL	SÍMBOL	BINARI	DECIMAL	SÍMBOL
010 0000	32	(Espai)	100 0000	64	@	110 0000	96	`
010 0001	33	!	100 0001	65	A	110 0001	97	a
010 0010	34	"	100 0010	66	B	110 0010	98	b
010 0011	35	#	100 0011	67	C	110 0011	99	c
010 0100	36	\$	100 0100	68	D	110 0100	100	d
010 0101	37	%	100 0101	69	E	110 0101	101	e
010 0110	38	&	100 0110	70	F	110 0110	102	f
010 0111	39	'	100 0111	71	G	110 0111	103	g
010 1000	40	(100 1000	72	H	110 1000	104	h
010 1001	41)	100 1001	73	I	110 1001	105	i
010 1010	42	*	100 1010	74	J	110 1010	106	j
010 1011	43	+	100 1011	75	K	110 1011	107	k
010 1100	44	,	100 1100	76	L	110 1100	108	l
010 1101	45	-	100 1101	77	M	110 1101	109	m
010 1110	46	.	100 1110	78	N	110 1110	110	n
010 1111	47	/	100 1111	79	O	110 1111	111	o
011 0000	48	0	101 0000	80	P	111 0000	112	p
011 0001	49	1	101 0001	81	Q	111 0001	113	q
011 0010	50	2	101 0010	82	R	111 0010	114	r
011 0011	51	3	101 0011	83	S	111 0011	115	s
011 0100	52	4	101 0100	84	T	111 0100	116	t
011 0101	53	5	101 0101	85	U	111 0101	117	u
011 0110	54	6	101 0110	86	V	111 0110	118	v
011 0111	55	7	101 0111	87	W	111 0111	119	w
011 1000	56	8	101 1000	88	X	111 1000	120	x
011 1001	57	9	101 1001	89	Y	111 1001	121	y
011 1010	58	:	101 1010	90	Z	111 1010	122	z
011 1011	59	;	101 1011	91	[111 1011	123	{
011 1100	60	<	101 1100	92	\	111 1100	124	
011 1101	61	=	101 1101	93]	111 1101	125	}
011 1110	62	>	101 1110	94	^	111 1110	126	~
011 1111	63	?	101 1111	95	_	111 1111	127	(Esborrar)

CODI BINARI I LLENGUATGES DE PROGRAMACIÓ

Tal com s'ha explicat, els ordinadors només poden entendre dues ordres: encès i apagat. De fet, el nucli dels ordinadors està format per bilions de transistors que funcionen com a interruptors, i tot allò que fa un ordinador es podria resumir en una combinació de transistors encesos i apagats. El codi binari s'utilitza per representar aquestes combinacions, on cada dígit representa un transistor (bit). El llenguatge màquina o codi màquina és el sistema d'instruccions i dades codificat en codi binari que poden entendre els microprocessadors.

Programar utilitzant directament aquest codi és inviable per la quantitat de temps que es necessitaria i, per aquest motiu, s'han desenvolupat diferents llenguatges de programació. Aquests ens permeten crear, per exemple, programes, aplicacions o webs utilitzant uns codis relativament senzills d'escriure, llegir i comprendre. Aquests llenguatges de programació duen integrats altres programes que tradueixen allò que escrivim a codi binari.

Segons com siguin de propers o distants els llenguatges de programació al codi binari, aquests es distingeixen en llenguatges de baix o d'alt nivell respectivament. Actualment, els llenguatges més utilitzats acostumen a ser l'alt nivell.

PENSAMENT COMPUTACIONAL

El pensament computacional és una manera de resoldre un determinat problema que integra la manera de «pensar» dels ordinadors amb els mecanismes de pensament humà. El pensament computacional busca solucions que es poden expressar com una seqüència d'instruccions o un algoritme. Si volem comunicar-nos amb els ordinadors, per exemple per programar-los, és important conèixer, a més del codi que utilitzen les màquines, quina és la lògica que segueixen.

Les seves característiques principals són:

- Formular problemes de manera que es puguin resoldre amb ordinadors.
- Organitzar les dades de manera lògica i analitzar-les.
- Representar les dades mitjançant abstraccions, com models i simulacions.
- Realitzar algoritmes per automatitzar solucions.
- Identificar, analitzar i implementar diferents solucions possibles amb l'objectiu de trobar la solució més eficient.
- Generalitzar i transferir el procediment de solució de problemes a d'altres de diferents.

Les actituds i habilitats que es desenvolupen treballant el pensament computacional són:

- Confiança en la capacitat d'enfrontar-se a problemes complexos.
- Persistència en el treball.
- Habilitat per treballar amb problemes poc estructurats.
- Habilitat per comunicar-se i treballar en equip.

El pensament computacional segueix sempre el mateix procediment, que consta de dues grans fases: la formulació del problema i la seva resolució, que alhora es divideix en quatre fases: descomposició, reconeixement de patrons, abstracció i creació de l'algoritme.

El primer pas per resoldre un problema satisfactòriament és **formular el problema**, és a dir, emmarcar i definir exactament quin és el problema que volem resoldre. Aquest pas pot semblar poc important o obvi, però és molt important, ja que, si no definim correctament el problema, la solució que plantegem no servirà per al propòsit real.

Un cop tenim el problema formulat, comença el procés per poder-lo resoldre. El primer que s'ha de fer és **descompondre el problema**, és a dir, fragmentar-lo en problemes més petits i senzills de resoldre. A continuació, cal **reconèixer patrons**, que són estructures que es repeteixen en les diferents parts. El pas següent és l'**abstracció**, que consisteix a detectar quines són les diferències entre les diferents parts i abstraure-les. Finalment, cal crear l'**algoritme**, és a dir, les instruccions que permetin solucionar el problema inicial.

Per entendre millor què són els algorismes i la seva importància en el nostre dia a dia, us recomanem l'article «El poder dels algorismes: com el software formata la cultura» publicat per Sandra Álvaro al blog del CCCB Lab.¹

CONCEPTES CLAU

- Programar
- Llenguatge (natural i formal)
- Codi
- Codi binari
- Pensament computacional / *Computational thinking*
- Definició de problemes / *Problem discovery*
- Resolució de problemes / *Problem solving*
- Descomposició
- Patró
- Algoritme

¹ ÀLVARO, Sandra (2014). «El poder dels algorismes: com el *software* formata la cultura» http://blogs.cccb.org/lab/article_el-poder-dels-algorismes-com-el-software-formata-la-cultura/



Un programador és idealment un assagista que treballa amb formes estètiques i literàries tradicionals, així com amb conceptes matemàtics, per comunicar la forma en què funciona un algorisme i convèncer un lector que els resultats seran correctes.

DONALD KNUTH



DINÀMICA DE L'ACTIVITAT

INTRODUCCIÓ: EL LABERINT

*Podem utilitzar el **Prezi** per conduir l'activitat*

Per tal de treballar la lògica de la programació, proposem un exercici de resolució de laberints utilitzant l'algoritme de Trémaux. Es recomana utilitzar laberints a escala humana dibuixats amb cinta, guix o pintura al pati, però si no també es pot treballar sobre paper.

En primer lloc, per entrar en situació i despertar la concentració dels alumnes, se'ls reparteix la fitxa amb laberints dibuixats (**Fitxa 1**) per tal que els resolguin de forma individual, sense donar-los cap instrucció precisa més enllà de que han de trobar el camí que porti a la sortida dels laberints.

Un cop hagin resolt els laberints, se'ls planteja la pregunta següent: com ho heu fet? Quina estratègia heu seguit? Probablement, la resposta sigui que han observat el laberint globalment i de seguida han vist quin era el camí a seguir. Aquí els podem anunciar que es pot seguir una estratègia per trobar la sortida a aquests laberints, però que la descobrirem més endavant.

LLENGUATGE I CODI: COMUNICAR-NOS AMB ELS ORDINADORS

A continuació es planteja el següent cas als alumnes: un de nosaltres acaba de ser introduït en un laberint com els que hem vist, i la resta som els encarregats de guiar-lo per a que surti del laberint, ja que ell no pot veure-hi més enllà del camí que té davant. Com ho podem fer? imagineu-vos que aquest company és un ordinador: com ens hi hauríem de comunicar? Quin llenguatge hauríem d'utilitzar? Quin tipus de codi?

A LLENGUATGE

En primer lloc, cal treballar quin tipus de llenguatge utilitzen els ordinadors. Per introduir els alumnes als conceptes de llenguatge natural i llenguatge formal es recomana realitzar les activitats «**Malentesos quotidians**» (Fitxa 3) i «**Dibuixa les instruccions**» (Fitxa 4). Amb aquests exercicis, els alumnes haurien d'entendre que el llenguatge que s'ha d'utilitzar per comunicar-se amb els ordinadors ha de ser molt precís, i que aquests no poden interpretar, és a dir, que executaran literalment les ordres que els donem.

MALENTESOS QUOTIDIANS (Fitxa 3)

Una de les característiques del llenguatge natural és que necessita ser interpretat. I és aquest procés, el de la interpretació, el que dona lloc als malentesos.

De la vida quotidiana a la programació:

Els malentesos es produeixen per diferents motius, o bé perquè la informació que s'ha transmès no era del tot precisa o perquè el missatge no s'ha interpretat seguint els mateixos criteris. Per tant, quan hi ha un malentès, el primer que cal fer és revisar si hi ha hagut un error d'interpretació.

Quines són les dues possibles interpretacions d'aquests textos?





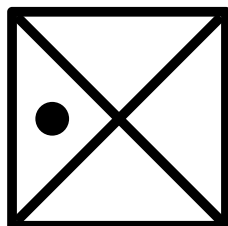
DIBUIXA LES INSTRUCCIONS (Fitxa 4)

Per fer palesa la diferència entre els dos tipus de llenguatge, es pot realitzar l'exercici següent. Es demana als alumnes que dibuixin en un full aquestes instruccions:

- Una línia recta horitzontal.
- Una línia recta vertical.
- Una altra línia recta horitzontal.
- Una altra línia recta vertical.
- Una X en el centre de l'estructura (si és que hem creat una estructura).
- Un punt a l'esquerra.

Una vegada s'hagi acabat de dibuixar, cadascú compara el seu dibuix amb el dels companys i s'adona que, en general, tots els dibuixos són diferents.

El que s'ha dictat volia representar aquest dibuix:



Per què ha passat això? No és culpa de no haver seguit les instruccions correctament, ha passat perquè els humans, quan parlem, interpretem. Cadascun de nosaltres ha interpretat les instruccions, que no eren prou precises, que s'han fet amb llenguatge natural.

Els programadors, quan volen escriure un programa, es troben amb problemes similars. D'una banda, han d'utilitzar un llenguatge que no doni lloc a confusions, ja que la màquina no interpretarà el missatge. D'altra banda, fins que no han acabat de donar les instruccions, no poden veure'n el resultat.

Una segona part de l'exercici podria consistir en què un alumne descrigui un altre dibuix similar a la resta de la classe, intentant afinar al màxim les instruccions per evitar malentesos. Donem un exemple de com es podria afinar la descripció anterior:

- Dibuixem una línia horitzontal.
- Dibuixem una línia vertical de la mateixa llargada que l'anterior, prenent com a punt inicial el final de la ratlla que ja havíem dibuixat i formant un angle recte.
- Dibuixem una línia horitzontal paral·lela a la primera i de la mateixa llargada i que parteixi del final de la ratlla anterior.
- Dibuixem una línia vertical que comenci al final de l'última línia dibuixada i que vagi fins a l'inici de la primera.
- Dibuixem dues línies diagonals que vagin de vèrtex a vèrtex de l'estructura dibuixada.
- Dibuixem un punt al centre del triangle que ha quedat format a la part esquerra de l'estructura.

Tot i intentar precisar al màxim les instruccions, veurem que sempre sorgeixen dubtes i que els dibuixos mai no seran exactament iguals que el de referència (mida exacta del punt, gruix de les línies, mida del quadrat...). Així doncs, si

haguéssim de donar aquestes instruccions a un ordinador, hauríem de refer-les i no deixar lloc a cap ambigüitat: mida i gruix exactes de les línies, mida del punt i situació exacta dins de l'estructura, color exacte...

B CODI

En segon lloc, cal estudiar quin codi utilitzarem per comunicar-nos amb un ordinador. Per treballar els codis es recomana realitzar les activitats «**Codis quotidians**» (Fitxa 5) i «**Codi binari**» (Fitxa 6). Aquestes haurien de servir perquè els alumnes comprenguin que un ordinador només pot entendre un codi binari que es correspongui a les dues accions més bàsiques que poden realitzar els seus components nuclears, els transistors.

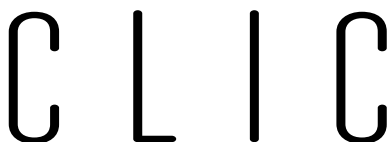
CODIS QUOTIDIANS (Fitxa 5)

Aquesta activitat ha de servir per reflexionar sobre què és un codi, quins són els codis que es coneixen i com estan construïts. Es parteix de l'anàlisi dels codis més populars per arribar a la qüestió dels codis que utilitzen els ordinadors.

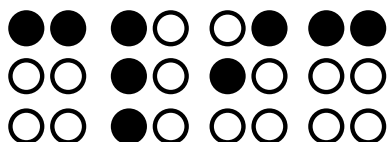
De la vida quotidiana a la programació:

Quins codis coneixem? Per a què serveixen? Com estan construïts? Quines variables els limiten? Què necessitem transmetre, de quines eines disposem? Què podem interpretar...

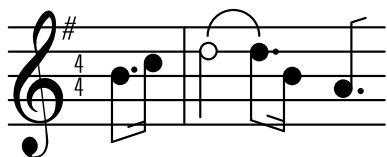
- Alfabet: llatí, braille... Per quins elements estan condicionats? Què diferencia fonamentalment l'alfabet braille del llatí? Per què?
- Codi de notació musical: què vol transmetre? Quines en són les variables? Com es combinen? Queda lloc per a la confusió o per a la interpretació? És positiu que quedi lloc per a la interpretació?
- Senyals de fum: quines en són les limitacions? Quins en són els recursos? Què es pot fer amb el foc? Quines variables podríem utilitzar?



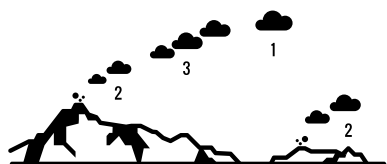
ALFABET LLATÍ



BRILLE



NOTACIÓ MUSICAL



SENYALS DE FUM

Què passa amb els ordinadors? Com ha de ser el seu codi? Les respostes a aquestes dues preguntes es troben a l'exercici següent.

CODI BINARI (Fitxa 6)

En aquest exercici s'explica el codi binari i per què aquest és l'utilitzat pels ordinadors. L'objectiu és entendre el sistema binari, crear un codi amb aquest sistema i posar-lo en funcionament.

En primer lloc, cal que els alumnes entenguin una sèrie de conceptes:

- Els ordinadors funcionen amb corrent elèctric, i aquest només té dos estats (obert i tancat); per tant, només poden rebre dues instruccions, una per l'obert i l'altra pel tancat. A aquests estats, doncs, se'ls ha assignat dos números, el 0 per a l'obert i l'1 per al tancat.
- Per tal d'enviar instruccions perquè facin determinades coses, hem d'establir un codi, és a dir, establir correspondències entre les instruccions que volem enviar i les instruccions que podem donar. En aquest punt, es pot demanar als alumnes que intentin crear un codi per poder escriure les lletres de l'abecedari. Poden fer servir aquesta taula (els números en vermell són de mostra).

UNITAT DIDÀCTICA # 2 – LA LòGICA DE LA PROGRAMACIÓ

EN 0 i 1	LLETRA	EN 0 i 1	LLETRA	EN 0 i 1	LLETRA
1	A	101	J	0011	S
0	B	011	K	0111	T
11	C	001	L	1111	U
00	Ç	000	M	0100	V
10	D	1111	N	0010	W
01	E	1110	Ñ	0110	X
111	F	1100	O	1011	Y
000	G	1000	P	1101	Z
110	H	0000	Q	1001	.
100	I	0001	R	0101	(espai)

Aquesta seria una manera de codificar, però veiem que té una complicació, i és que, com que la successió de 0 i 1 és arbitrària, és difícil sistematitzar-ho. Una solució per posar ordre a aquest caos és aprofitar la lògica del sistema binari.

El sistema binari segueix un sistema de numeració de base 2 que només té dos símbols, el 0 i l'1. Cal multiplicar el 0 o l'1 que hi hagi per 2 elevat a l'exponent que correspongui: començant de dreta a esquerra, cada posició té un exponent que supera una unitat a l'anterior, sent 0 el primer de tots. Finalment, caldrà sumar tots els resultats i obtindrem el número en sistema decimal.

Exemple:

$$1100 = 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 = 0 + 0 + 4 + 8 = 12$$

$$1001 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 1 + 0 + 0 + 8 = 9$$

Taula per a l'exercici:

$2^4 = 16$		$2^3 = 8$		$2^2 = 4$		$2^1 = 2$		$2^0 = 1$		SISTEMA DECIMAL
0	+	0	+	1	+	1	+	1	=	7
1	+	0	+	0	+	1	+	1	=	19
1	+	1	+	0	+	1	+	1	=	27

UNITAT DIDÀCTICA # 2 – LA LòGICA DE LA PROGRAMACIÓ

Amb aquest sistema, obtenim una doble correlació, entre el número binari, el decimal i el símbol que volem representar. Ara la taula quedaria de la manera següent:

DECIMAL	BINARI	LLETRA	DECIMAL	BINARI	LLETRA	DECIMAL	BINARI	LLETRA
0	0	A	10	1010	J	20	10100	S
1	1	B	11	1011	K	21	10101	T
2	10	C	12	1100	L	22	10110	U
3	11	Ç	13	1101	M	23	10111	V
4	100	D	14	1110	N	24	11000	W
5	101	E	15	1111	Ñ	25	11001	X
6	110	F	16	10000	O	26	11010	Y
7	111	G	17	10001	P	27	11011	Z
8	1000	H	18	10010	Q	28	11100	.
9	1001	I	19	10011	R	29	11101	(espai)

I, per simplificar-ho més encara, podem fer que totes les lletres tinguin la mateixa llargària de dígits, i per aconseguir-ho simplement hem d'afegir un 0 al davant fins a arribar a cinc xifres; així:

DECIMAL	BINARI	LLETRA	DECIMAL	BINARI	LLETRA	DECIMAL	BINARI	LLETRA
0	00000	A	10	01010	J	20	10100	S
1	00001	B	11	01011	K	21	10101	T
2	00010	C	12	01100	L	22	10110	U
3	00011	Ç	13	01101	M	23	10111	V
4	00100	D	14	01110	N	24	11000	W
5	00101	E	15	01111	Ñ	25	11001	X
6	00110	F	16	10000	O	26	11010	Y
7	00111	G	17	10001	P	27	11011	Z
8	01000	H	18	10010	Q	28	11100	.
9	01001	I	19	10011	R	29	11101	(espai)

UNITAT DIDÀCTICA # 2 – LA LòGICA DE LA PROGRAMACIÓ

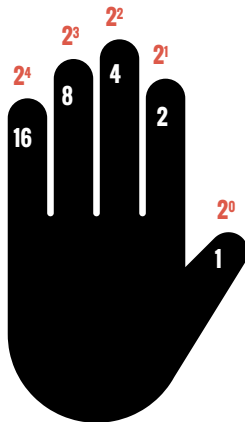
I veiem que encara hi podríem afegir dos símbols més, corresponents als números 30 i 31:

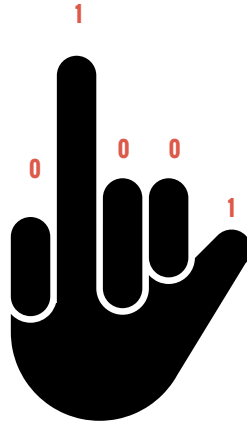
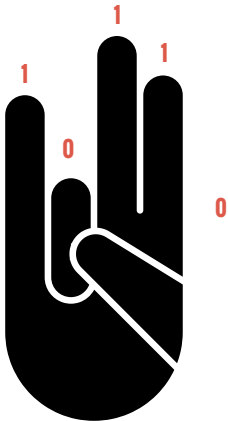
DECIMAL	BINARI	LLETRA
30	11110	
31	11111	

Una vegada tenim el codi establert, podem jugar a codificar paraules o frases. Es proposen dos exercicis diferents:

Exercici A: Un alumne escriu seqüències de cinc díigits en un paper i un altre les ha de descodificar fent servir la taula de conversió que s'ha creat de manera col·lectiva.

Exercici B: Exercici amb els dits de la mà. Els cinc dits de la mà representen els cinc díigits que equivalen a cada símbol del codi creat. L'exemple s'ha creat utilitzant la mà dreta per transmetre el codi. Tal com es pot veure, a cada dit li correspon una posició del nombre. El dit polze equival a la primera posició començant pel darrere. Un dit aixecat representa un 1, i un dit amagat representa un 0. Els alumnes es poden escriure als dits amb retolador les potències o els números (com en el dibuix) per facilitar la conversió al sistema decimal. En els dibuixos es poden veure exemples de lletres transmises amb aquest sistema, basat en la taula de conversió de més amunt.





10110 = 22 = U

01001 = 9 = I

Un cop finalitzada l'activitat, es pot ensenyar als alumnes la taula ASCII.

Versió simplificada:

A	■	□	■	■	■	■	■	■
B	■	□	■	■	■	■	■	■
C	■	□	■	■	■	■	■	■
D	■	□	■	■	■	■	■	■
E	■	□	■	■	■	■	■	■
F	■	□	■	■	■	■	■	■
G	■	□	■	■	■	■	■	■
H	■	□	■	■	■	■	■	■
I	■	□	■	■	■	■	■	■
J	■	□	■	■	■	■	■	■
K	■	□	■	■	■	■	■	■
L	■	□	■	■	■	■	■	■
M	■	□	■	■	■	■	■	■

N	■	□	■	■	□	□	□	■
O	■	□	■	■	□	□	□	■
P	■	□	■	■	■	■	■	■
Q	■	□	■	■	■	■	■	■
R	■	□	■	■	■	■	■	■
S	■	□	■	■	■	■	■	■
T	■	□	■	■	■	■	■	■
U	■	□	■	■	■	■	■	■
V	■	□	■	■	■	■	■	■
W	■	□	■	■	■	■	■	■
X	■	□	■	■	■	■	■	■
Y	■	□	■	■	■	■	■	■
Z	■	□	■	■	■	■	■	■

Amb la taula de conversió del codi ASCII extens Latin-1, codificar i descodificar diferents missatges. Per fer aquest exercici, es pot utilitzar la taula de l'ASCII o una versió encara més simplificada on el número binari es representa amb seqüències de vuit quadradets pintats, on el blanc representa el 0 i el negre l'1.

C CODI LABERINT (Fitxa 7)

Arribats a aquest punt, els alumnes haurien d'establir un codi utilitzant el sistema binari per a transmetre les ordres bàsiques al seu company.

Per fer-ho més versemblant, es recomana dibuixar al pati els laberints (Fitxa 2) a escala humana amb cinta aïllant, guix o pintura. D'aquesta manera, encara que no hi hagi parets que limitin la visibilitat, serà més difícil que els alumnes trobin la sortida des de dins del laberint. Si no es poden dibuixar els laberints a gran escala, una altra opció és imprimir aquests laberints en A3 i que es faci el recorregut amb els dits o desplaçant un ninot tipus Clic.

La dinàmica és la següent:

- Els alumnes es divideixen en grups de quatre.
- Es reparteix una còpia de la fitxa 7 a cada grup.
- S'escull un representant de cada grup, que serà el que haurà d'entrar al laberint. Aquest surt fora de l'aula.
- Els altres tres membres s'han d'inventar un codi per transmetre les ordres del laberint. Les ordres que han de transmetre són: avança, atura't, gira a la dreta 90° i gira a l'esquerra 90°. Per fer-ho, han d'escollir 4 números, escriure'ls també en binari i omplir la fitxa. L'opció més senzilla és fer-ho amb números petits, de l'1 al 4, per exemple. Però es pot complicar l'activitat si els obliguem a que assignin números més grans, entre el 8 i el 15, a cada ordre.
- Quan els grups hàgin creat el seu codi, han de retallar la fitxa per la línia de punts, quedar-se la part A i entregar al seu representant la part B (que no podrà veure mai la part A).
- A continuació, els equips competeixen per guiar en el menor temps possible el seu representant fins a la sortida del laberint utilitzant el codi que han creat. És molt important que els representants segueixin al peu de la lletra les instruccions dels seus companys perquè si no poden ser penalitzats i obligats a començar de nou el recorregut.

Exemple de fitxa:

Part A

BINARI	BINARI	ORDRE
1	001	Avança
2	010	Atura't
3	011	Gira a la dreta 90°
4	100	Gira a l'esquerra 90°

DECIMAL	ORDRE
1	Avança
2	Atura't
3	Gira a la dreta 90°
4	Gira a l'esquerra 90°

Part B

BINARI	ORDRE
011	Gira a la dreta 90°
010	Atura't
100	Gira a l'esquerra 90°
001	Avança

D ALGORITMES: SORTIR DEL LABERINT

Una vegada els alumnes hagin jugat a donar instruccions al seu company per sortir del laberint, s'introdueix una nova variable: hi ha tres laberints diferents (**Fitxa 2**) i no sabem en quin laberint ens tancaran. Per tant, hem d'estudiar els tres laberints i descobrir l'estratègia que ens permeti sortir de qualsevol d'ells. No s'hi val a memoritzar els camins, perquè pot ser que ens introdueixin en un quart laberint que no coneixem però del qual en podrem sortir utilitzant la mateixa estratègia.

La solució a aquest exercici consisteix en utilitzar l'algoritme de Trémaux. És probable que la majoria d'alumnes no aconseguixin trobar-la, així que podem optar, o bé per donar-los la solució, o bé per anar-los donant pistes fins que, com a mínim, algun alumne ho resolgui.

L'algoritme de Trémaux va ser creat per un enginyer francès de finals del segle XIX, Charles Trémaux, per trobar la sortida de qualsevol laberint.

L'ús d'aquest algoritme garantitza passar per tot el recorregut del laberint. D'aquesta manera, tard o d'hora, es trobarà la sortida, i, si no n'hi ha, ens tornarà al punt de partida.

Les instruccions d'aquest algoritme són les següents:

1. No transitis el mateix camí dues vegades en el mateix sentit.
2. Si arribes a una cruïlla nova, agafa qualsevol camí.
3. Si un camí nou porta a una cruïlla vella o a un carreró sense sortida, retrocedeix fins a la cruïlla anterior.
4. Si un camí vell porta a una cruïlla vella, agafa un camí nou, i, si no en queden, agafa'n qualsevol.

Per posar-lo en pràctica haurem de tenir en compte dues coses:

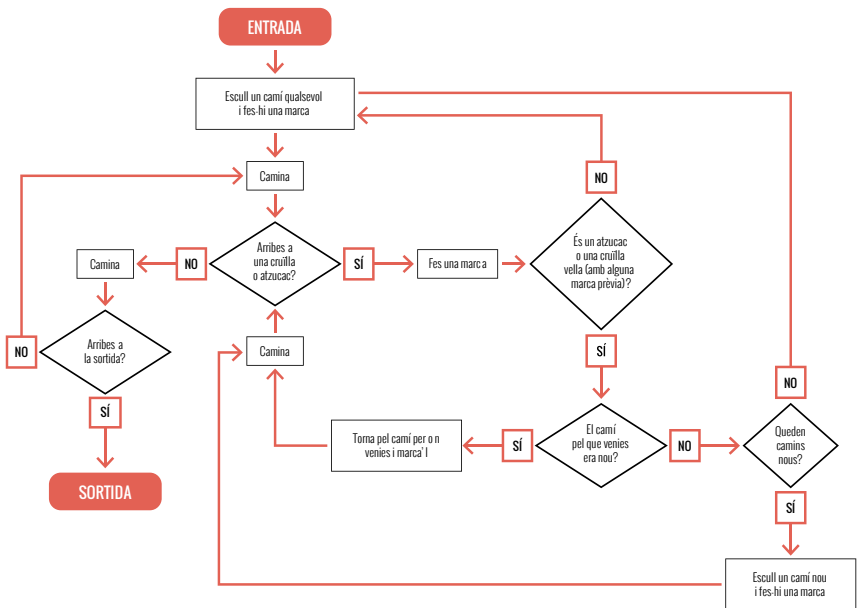
- Sempre hem de saber si estem transitant un camí per primera o segona vegada.
- Sempre que agafem un camí o arribem a una cruïlla, hem de deixar-hi una marca. Així quan arribem a un encreuament, sempre sabrem si els camins que veiem són nous (si no tenen cap marca), si els hem transitat una vegada (si tenen una marca) o si ja els hem transitat dues vegades i no els podem tornar a agafar (si tenen dues marques)

UNITAT DIDÀCTICA # 2 – LA LòGICA DE LA PROGRAMACIÓ

Quan s'hagi explicat l'algoritme es demana als alumnes que creïn un diagrama de flux que representi l'algoritme de Trémaux (**Fitxa 8**). A partir d'aquí, ja poden comprovar si funciona en els laberints dibuixats al terra o ens els laberints impresos. Per marcar el pas per les cruïlles, es poden utilitzar, per exemple, pedres o cigrons.

Per fer-ho més emocionant es poden organitzar curses cronometrades pels diferents laberints.

El diagrama de flux hauria de quedar així:



TANCAMENT DE L'ACTIVITAT

Per tancar l'activitat de manera distesa, es proposa veure conjuntament una escena de la sèrie «The Big Bang Theory». Els protagonistes de la sèrie són uns científics d'elit, molt intel·ligents. Un d'ells, en Sheldon, té una manera força peculiar de relacionar-se. A l'escena que es veurà, en Sheldon ha decidit crear un algoritme per fer amics i el presenta als seus companys:

<http://youtu.be/LT6cyqPQq2E>

Després de veure el vídeo, ens podríem plantejar les preguntes següents:

- Està ben construït l'algoritme?
- Per què no funciona? Soluciona el problema real d'en Sheldon?

Podem analitzar l'algoritme i veurem que no té cap problema formal, però aquest no funciona perquè no soluciona el problema real d'en Sheldon. Amb aquest algoritme potser aconseguirà organitzar una activitat amb algú, però no haurà resolt els seus problemes de relació.

A partir d'aquí, es pot proposar als alumnes que s'imaginin qualsevol problema i creïn un algoritme per resoldre'ls. Després es poden posar en comú i comprovar-ne la seva funcionalitat.

PROPOSTA D'AVALUACIÓ DE L'ACTIVITAT

Es proposa que, per avaluar la unitat didàctica, es realitzi una documentació i síntesi de l'activitat realitzada. El document final hauria de recollir:

- Imatges (estàtiques o en moviment) del procés.
- Un resum d'allò que s'ha treballat i com s'ha fet.
- Els dubtes principals que han sorgit.
- Les propostes creatives que hagin sorgit a partir de l'activitat.
- Una valoració de l'activitat. Seria interessant que recollís tant l'opinió del professor com la dels alumnes. Es pot fer una valoració global i incloure-hi citacions individuals.
- Propostes de millora o recomanacions per a altres grups que decideixin dur a terme l'activitat.

Amb aquest exercici es vol aconseguir:

- Treballar continguts i competències transversals: processament de la informació, comunicació i expressió, capacitat de síntesi i treball en equip.
- Compartir el coneixement adquirit.
- Donar visibilitat al treball realitzat.

Es proposa que els alumnes treballin per comissions (fotografia, vídeo, redacció, correcció, edició...) i que, un cop l'exercici estigui finalitzat, es comparteixi a través de la plataforma del CCCB Educació (www.cccbeducació.org).

ORIENTACIONS METODOLÒGIQUES

Les activitats s'agrupen en tres blocs: llenguatge, codi i pensament computacional. Els tres blocs segueixen la mateixa estructura metodològica: es parteix d'una activitat centrada a analitzar alguna situació de la vida quotidiana i es provoca, a través de la formulació de preguntes, que siguin els mateixos alumnes els que n'extreguin els conceptes principals. A continuació, es presenta com a mínim una activitat que obliga a generalitzar el coneixement adquirit a la primera per aplicar-lo en una situació més abstracta.

Aquesta metodologia vol donar eines als estudiants per poder analitzar el seu entorn més proper a fi d'extreure'n coneixement, és a dir, els dóna eines per aprendre a aprendre.

Com a activitat d'avaluació, se suggereix documentar la sessió, seguint un doble objectiu: processar el coneixement adquirit i compartir el treball realitzat.

MATERIAL COMPLEMENTARI



Web



Article



Vídeo



Llibre



Conferència



Arxiu de la segona sessió de formació d'**Univers Internet**:

- Vídeo de la xerrada: Internet ens fa més superficials o més llestos? Cristina Sáez.
http://www.cccb.org/ca/video-univers_internet_2_persona_cristina_saez_vo_es-45516
- Vídeo del taller: La lògica de la programació: de la vida quotidiana a la programació. Irene Lapuente.
http://www.cccb.org/ca/video-univers_internet_2_persona_irene_lapuente_vo_es-45517
- Articles publicats al blog del CCCB Lab:
 - Univers Internet, més superficials o més llestos? Cristina Sáez.
http://blogs.cccb.org/lab/article_univers-internet-mes-superficials-o-mes-llestos/
 - Tecnorevolució a les aules. Cristina Sáez.
http://blogs.cccb.org/lab/article_tecnorevolucio-a-les-aules/














Web de **La Mandarina de Newton**: www.lamandinadenewton.com/ca








ALBA, Rafael. «Aprender a programar... ¿Desde pequeños?»
<http://recursostic.educacion.es/observatorio/web/ca/software/programacion/593-rafael-alba> A quina edat s'ha de començar a ensenyar a programar i quines són les eines més indicades en cada moment?



ÁLVARO, Sandra (2014). «El poder dels algorismes: com el software formata la cultura» http://blogs.cccb.org/lab/article_el-poder-dels-algoritmes-com-el-software-formata-la-cultura/

-  BEJERANO, Pablo G. «Aprender a programar es el futuro de la educación primaria» <http://blogthinkbig.com/aprender-programar-futuro-educacion/>. En aquest breu article es presenten els beneficis d'ensenyar a programar, acompanyats d'exemples d'èxit i d'eines útils.
-  «Code Academy» <http://www.codecademy.com/es> . Acadèmia virtual de referència que ensenya a programar amb múltiples recursos.
-  «Computer Science... without a Computer!» <http://csunplugged.org/> Recursos per treballar la programació sense ordinadors. La guia didàctica està traduïda al castellà.
-  «edX» <https://www.edx.org/>. Plataforma que ofereix cursos virtuals de les millors universitats del món i fomenta l'aprenentatge autònom.
-  «Exploring Computational Thinking» <http://www.google.com/edu/computational-thinking/> Eina desenvolupada per Google per desenvolupar les capacitats del pensament computacional.
-  «Kodu» <http://www.kodugamelab.com/>. Eina per programar jocs sense coneixements de programació.
-  «Mozilla Foundation» <http://www.mozilla.org/en-US/foundation/>. Web de la Fundació Mozilla, que té com a objectiu convertir els seus usuaris en creadors de webs.
-  «MIT App Inventor» <http://appinventor.mit.edu/explore/>. Eina desenvolupada pel MIT per aprendre a programar aplicacions i recursos per dur-los a terme.
-  OLSON, Pamy (2012). «Why Estonia Has Started Teaching Its First-Graders To Code» <http://www.forbes.com/sites/pamyolson/2012/09/06/why-estonia-has-started-teaching-its-first-graders-to-code/> A Estònia s'ha introduït la programació en el currículum escolar amb molt bons resultants. Per què ho han fet i quins han estat els resultants obtinguts?
-  PASTOR, Javier. «Cinco motivos por los que los niños deben aprender a programar» <http://mobileworldcapital.com/es/articulo/262> Aquest article dóna cinc arguments per defensar la necessitat d'ensenyar els nens a programar.
-  RESNICK, Mitch (2012). «Enseñemos a los niños a codificar» http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code.html Resnick descriu els beneficis d'ensenyar els nens a programar perquè no només puguin llegir les noves tecnologies, sinó també crear-les.

-  «Scratch» <http://scratch.mit.edu/>. Projecte del MIT. Una eina per programar de manera força intuïtiva, apta per a nens i adults.
-  SILIÓ, Elisa (2013). «Aprender a programar como se aprende a leer» http://sociedad.elpais.com/sociedad/2013/03/07/actualidad/1362689630_904553.html. Article que planteja la necessitat d'ensenyar a programar a l'escola.
-  «The Hour of Code». <http://csedweek.org/learn> Tutorials per aprendre a programar.
-  «Thinkersmith». <http://thinkersmith.org> . Web amb múltiples recursos per apropar la tecnologia als nens de manera saludable i per ensenyar a programar.
-  WILLIAMSON, Ben (2013). «Programming Power? Does Learning to Code Empower Kids?» <http://dmlcentral.net/blog/ben-williamson/programming-power-does-learning-code-empower-kids> Una visió política de què significa apoderar a través d'ensenyar a programar.



Màquina Colossu.
Primer ordinador programable (1943).
Font: Viquipèdia.

Univers Internet

Com afecta la Xarxa les nostres vides?

Maleta pedagògica

www.cccbeducacio.org/universinternet

CCCB Educació / CCCB Lab

Aquest document està sota la llicència:

