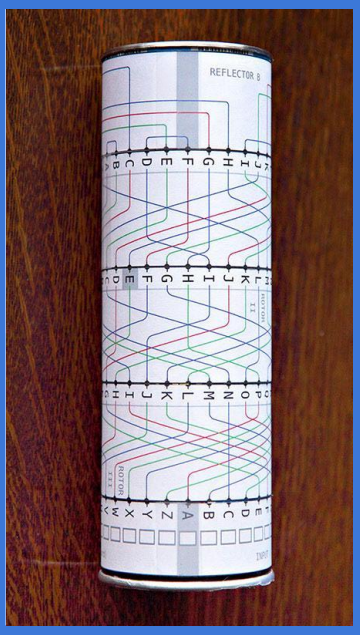
# CRIPTOGRAFIA: CREACIÓ D'UN PROGRAMA D'ENCRIPTACIÓ I DESENCRIPTACIÓ AMB SCRATCH




Mireia Marcos Cañal

Grup: 4A

Tutor: Jaime Morcillo García

Data de lliurament: 9 - 04 - 2021

**ÍNDEX**

## 1. INTRODUCCIÓ

El tema que tracto en aquest treball, com el seu títol indica, és l'estudi de la criptografia i la seva aplicació en la transmissió de missatges encriptats. La criptografia es basa en l'escriptura d'informació amb una clau secreta o de manera enigmàtica. A la comunicació, hi ha el paper de l'emissor, que és qui transmet el missatge, i el receptor, que és qui el rep. Malgrat això, també pot haver-hi una tercera persona l'objectiu del qual és conèixer el missatge, i la criptografia serveix per protegir aquesta informació.

La criptografia forma part de la criptologia, que és una disciplina que estudia les escriptures secretes, i per a aconseguir que l'encriptació o codificació d'informació sigui més eficaç, s'utilitzen i desenvolupen tècniques i procediments matemàtics. La criptografia és especialment important a l'àmbit matemàtic i informàtic, però també per a les activitats diàries de tothom.

La meva motivació per a realitzar el meu projecte de recerca d'aquest tema va sorgir pel meu gust per les matemàtiques. Des de sempre m'havien interessat, però vaig acabar de decantar-me per la criptografia gràcies al programa matemàtic Estalmat. El nom d'aquest programa està format per les sigles de *estímul del talent matemàtic*, i és un projecte de detecció i estímul del talent precoç a les matemàtiques que es dona en joves de dotze a quinze anys i que dura dos anys. Fa un any, vaig aconseguir formar part d'aquest programa, i la conferència inaugural d'aquest curs, impartida per Joan Jareño, va tractar sobre la criptografia. A més, una de les sessions a les que vaig assistir aquest any va ser una introducció a la criptografia, on els matemàtics Jordi Deulofeu i Abraham de la Fuente van ampliar alguns coneixements ja exposats a l'acte d'inauguració, i com que el tema m'interessava molt, vaig decidir dur a terme el meu projecte de recerca sobre la criptografia.

L'objectiu principal d'aquest treball és la creació d'un programa d'encriptació i desencriptació amb alguna aplicació informàtica coneguda per mi, com l'Scratch, i la comparació del programa creat amb els diferents mètodes

criptogràfics que han existit al llarg de la història per tal d'avaluar la seva eficàcia.

## 2. HIPÒTESIS I OBJECTIUS

Les hipòtesis que m'he plantejat per a començar a fer la recerca d'aquest projecte són les següents:

**És possible realitzar un programa d'encriptació i desencriptació amb una aplicació informàtica semblant a la màquina Enigma.**

**L'encriptació de missatges mitjançant aplicacions informàtiques és més efectiva que l'encriptació de missatges manual amb mètodes mecànics.**

Per contrastar aquesta hipòtesi hauré de realitzar una recerca històrica sobre la criptografia i els mètodes d'encriptació, per una banda, i sobre els instruments d'encriptació, centrant-me en la màquina enigma, per una altra.

Posteriorment realitzaré la meva part pràctica, que consistirà en dur a terme l'encriptació i desencriptació d'un missatge mitjançant un programa informàtic prèviament ja conegut per mi.

Durant la realització del treball, i per tal d'arribar a validar o refutar les meves hipòtesis em plantejaré els objectius següents:

- Realitzar l'encriptació i desencriptació d'un missatge mitjançant un mètode manual o mecànic.
- Realitzar un programa informàtic amb l'aplicació Scratch que em permeti encriptar i desencriptar el mateix missatge.
- Comparar ambdós models, examinant-ne els avantatges i desavantatges de la utilització d'un i altre.

Els objectius implícits del treball, i que aniré desenvolupant al llarg del mateix són els següents:

- Conèixer la història de la criptografia.
- Estudiar instruments de criptografia, com la màquina Enigma.
- Aprendre a programar a nivell avançat amb Scratch.
- Realitzar proves d'encriptació i desencriptació de missatges.

# 3. LA CRIPTOGRAFÍA

## 3.1. QUÈ ÉS LA CRIPTOGRAFIA

La criptografia és la ciència i l'art de transmetre missatges de manera xifrada o mitjançant un codi, i que tracta de transmetre aquesta informació protegint-la d'observadors no autoritzats. La paraula "criptografia" prové del grec, dels mots *kryptós* (*κρύπτος*), que significa ocult, i *graphé* (γραφή), que significa escriure.

Gràcies a l'evolució de la tecnologia, les diferents tècniques criptogràfiques han estat àmpliament divulgades i millorades mitjançant algorismes matemàtics. A més de protegir la seguretat de la informació transmesa, també proporciona seguretat a la identitat de l'usuari emisor i el receptor.

La criptografia pertany a un àmbit d'estudis, la criptologia, que tracta les comunicacions secretes per a diverses finalitats, com autentificar la identitat dels usuaris, autentificar i protegir les comunicacions personals, les transaccions comercials i bancàries, i protegir la integritat de les transferències electròniques de fons.

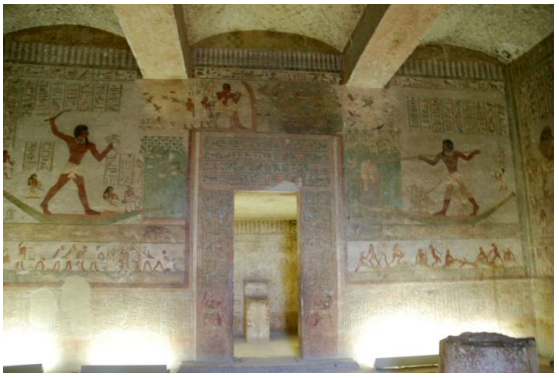
L'objectiu de la criptografia és dissenyar i aplicar sistemes criptogràfics per proporcionar seguretat, i el tipus de propietats de les quals s'encarrega són:

- Garantir la confidencialitat, és a dir, fer que la informació sigui accessible només per a personal autoritzat mitjançant codis i tècniques de xifrat.
- Proporcionar integritat, és a dir, garantir la correcció de la informació.
- Vincular un document, transacció o informació a una persona o un sistema de gestió criptogràfic autoritzat.
- Proporcionar una autenticació i aportar mecanismes que permetin verificar la identitat del transmissor del missatge.

## 3.2. HISTÒRIA DE LA CRIPTOGRAFIA

La criptografia, encara que és molt utilitzada a l'actualitat, té uns orígens molt remots. Hi ha constància que ja existia a les civilitzacions més antigues, ja que n'hi ha evidències tant a les escriptures mesopotàmiques com a les egípcies.

L'exemple més antic es va trobar a la tomba d'un noble egipci, Khnumhotep II, que va viure fa 3900 anys aproximadament. S'havien intercanviat les lletres de la inscripció per símbols, però en aquest cas no era per protegir informació, sinó per augmentar el seu atractiu lingüístic. Malgrat això, es considera que va començar a establir els principis de la criptografia. El primer cas de criptografia dedicada a ocultar missatges es va donar va 3500 anys, ja que un escriba mesopotàmic va tractar de protegir la fórmula del glacejat d'una ceràmica.



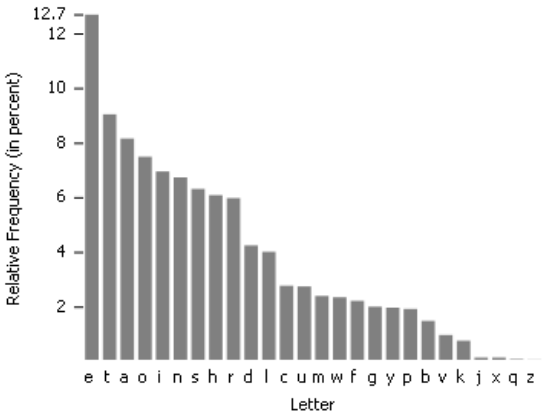
**Imatge 1.** Tomba del faraó Khnumhotep II. *Imatge extreta de http://amigosdelantiguoegipto.com/?page_id=3861*

Posteriorment, la criptografia va ser majoritàriament utilitzada per a ocultar informacions militars. En primer lloc, la ciutat d'Esparta, els missatges s'encriptaven col·locant un pergamí amb la informació enroscat en un cilindre d'una mesura particular. Es necessitava el diàmetre exacte del cilindre per poder llegir el missatge, ja que si no, la tira de paper només contenia lletres arbitràries. Els espies de l'antiga Índia també utilitzaven mètodes similars el segle II a.C.

Malgrat això, es considera que la criptografia antiga més avançada era la dels romans. Per exemple, van ser ells els primers en crear el xifratge de Cèsar, que consistia en assignar a cada lletra de l'abecedari una altra lletra, seguint un patró relacionat amb la posició de les lletres. Al receptor només li calia saber el nombre de posicions que havien de desplaçar-se les lletres per obtenir el nou abecedari, amb la qual cosa era un mètode bastant pràctic de codificar i

descodificar missatges. Aquest tipus de xifrat era per substitució, ja que a cada lletra li assignaven un nou valor.

Durant l'Edat Mitjana, encara que la criptografia va anar adquirint un paper més important, van continuar utilitzant-se mètodes d'encriptació per substitució, com el del xifratge de Cèsar. Malgrat això, al voltant de l'any 800 d.C., un matemàtic àrab anomenat Al-Kindi va crear una tècnica, l'anàlisi de freqüència, que vulnerava la seguretat de tots els sistemes criptogràfics de substitució. Aquesta tècnica es basava en analitzar la freqüència amb què s'utilitzaven totes les lletres de l'abecedari en diferents llengües, i comparar-la amb la freqüència de l'ús de les lletres al text xifrat. Que les freqüències de dues lletres iguals coincidissin significava que una d'aquestes lletres estava substituint l'altra. Això va suposar un gran problema per a la criptografia, i va haver de continuar evolucionant per no caure en desús.

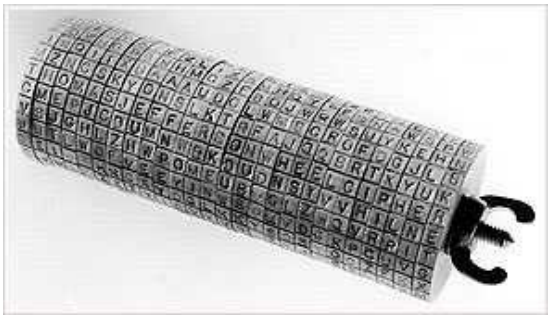



**Imatge 2.** Anàlisi de freqüència. *Imatge extreta de http://www.sinfocol.org/2009/06/crypto-badness-100-analisis-de-frecuencias/*

L'any 1465, Leone Alberti va crear el mètode d'encriptació per xifrat polialfabètic, és a dir, en comptes de canviar les lletres mitjançant un abecedari, es fa amb més d'un. Per tant, va crear una combinació dels mètodes de xifrat ja coneguts que incrementava considerablement la seguretat del missatge ocult. A més, també es van crear nous mètodes de xifrat durant el Renaixement, incloent-hi el mètode de codificació amb llenguatge binari o Codi Bacon, inventat per Francis Bacon. Amb aquesta tècnica, cada lletra se substitueix per un grup de cinc lletres, que poden ser la "A" o la "B", i a partir d'aquí es comencen a fer les frases. A més, Francis Bacon va crear i exposar les tres propietats que cada bon sistema de xifrat hauria de tenir:

- Ha de ser fàcil d'escriure i de llegir.
- Ha de ser confiable i que no pugui ser desxifrable.

- Si és possible, ha d'estar lliure de sospita.



**Imatge 3.** Roda de xifrat. *Imatge extreta de http://criptografiaurjc.blogspot.com/2008/04/el-cilindro-de-jefferson.html*

L'any 1790 va tenir lloc un important fet per a l'evolució de la criptografia: Thomas Jefferson va crear la roda de xifrat. En aquest cas, s'utilitzaven 36 discs que tenien escrit l'abecedari en diferents ordres. Després, es col·locaven en un cilindre, i es començaven a girar per tal d'obtenir a una línia el missatge desitjat. Per últim, només calia escollir una altra línia, i el resultat seria el missatge xifrat. Malgrat això, la roda de xifrat mai no va ser construïda en aquell moment, però sí que va utilitzar-se posteriorment.

Després, amb la Segona Guerra Mundial va aparèixer l'aplicació perfecta de tots els sistemes criptogràfics coneguts: es va inventar la màquina Enigma. Aquesta màquina utilitzava la idea de Jefferson de la roda de xifrat, i, encara que tenia el mateix aspecte que una màquina d'escriure, els seus mecanismes interns eren veritablement complexos. Mitjançant unes rodes giratòries, aconseguia codificar un missatge, i es van necessitar molts anys per poder trencar el xifrat d'Enigma.

Actualment, gràcies a l'aparició dels ordinadors, la criptografia ha començat a avançar molt més ràpidament. Per exemple, han aparegut nous mètodes criptogràfics, com l'encriptació matemàtica de 120-bits, que utilitzen molts dispositius i programes informàtics, o el sistema RSA, que s'empra amb el comerç electrònic.

Es pot veure com la criptografia ha recorregut un llarg camí en 4000 anys, però encara no l'ha acabat. Si continua necessitant-se protegir informació, la criptografia continuarà evolucionant, i apareixeran nous mètodes per xifrar cada vegada més sofisticats i segurs.

## 3.3. TIPUS DE SISTEMES CRIPTOGRÀFICS

Els primers mètodes per ocultar informació són molt antics i diversos, tal i com s'ha explicat prèviament: per exemple, a les antigues civilitzacions afaitaven el cabell d'un esclau, escrivien el missatge al seu cap, esperaven que li creixés i després l'enviaven al receptor del missatge. Una altra tècnica, com es va trobar a les tombes de l'antic Egipte, consistia en utilitzar signes equivalents a lletres o paraules i transmetre els signes.

Actualment, dintre de la criptografia en podem distingir dos mètodes diferents:

- Transposició: consisteix en canviar les posicions de les lletres d'un missatge, però aquestes continuen sent les mateixes.
- Substitució: conserva la posició de les lletres, però les canvia o substitueix per unes altres.

Al llarg de la història s'han creat molts sistemes criptogràfics cada vegada més eficients, hi alguns dels més remarcables són els següents:

- Dents de serra: aquest tipus de xifrat segueix el mètode de transposició, i consisteix en el següent: per exemple, suposem que tenim un missatge. En aquest cas serà "setze jutges d'un jutjat mengen fetge d'un penjat". A continuació haurem d'escriure el missatge per línies de la manera següent:

  S T E U G S U J T A M N E F T E U P N A
  E Z J T E D N U J T E G N E G D N E J T

  Finalment, enviaríem el missatge escrivint una línia a continuació de l'altra:

  S T E U G S U J T A M N E F T E U P N A E Z J T E D N U J T E G N E G D N E J T

  Per desxifrar el missatge, només caldria invertir el procés. Això sí, el receptor del missatge necessitaria conèixer el nombre de línies que

s'han emprat per codificar el missatge, ja que en comptes d'escriure el missatge en dues línies com a l'exemple, s'hauria pogut fer en més. Malgrat això, aquesta tècnica no és massa segura per encriptar missatges, ja que només caldria anar provant amb diferents nombres de línies per desxifrar el missatge si es coneix que la tècnica utilitzada per encriptar és la de dents de serra.

- Mètode dels espartans: el primer aparell conegut que fa servir la transposició va ser usat en el xifrat de missatges pels governants espartans cap a l'any 400 a.C. Com ja es va explicar prèviament, els missatges s'encriptaven de la següent manera: es feien dos bastons idèntics, amb el mateix diàmetre, un per al receptor del missatge i un per a l'emissor. El missatge s'escrivia en un tros de pergamí enrotllat en un dels bastons, i un cop desenvoltat, el missatge no es podia llegir, ja que semblava que la tira de paper només contenia lletres, sense seguir cap ordre.


**Imatge 4.** Mètode dels espartans. *Imatge extreta de https://blogs.salleurl.edu/es/networking-and-internet-technologies/cifrado-de-datos-imprescindible*

- Mètode de les columnes: és un altre exemple del mètode de transposició, i procedeix de la manera següent: en primer lloc, s'ha d'escollir una paraula clau, que en aquest cas serà "mates". Després, hem de col·locar el nostre missatge (que en el meu cas serà "mètode de les columnes") i la paraula a unes columnes, i el nombre de columnes estarà determinat pel nombre de lletres de la paraula clau. En aquest cas, hauríem de procedir de la manera següent:

| paraula clau → | **M** | **A** | **T** | **E** | **S** |
|---|---|---|---|---|---|
| missatge (s'ha de tenir en compte | m | e | t | o | d |

| | | | | | |
|---|---|---|---|---|---|
| que si sobren espais, tal i com és el cas d'aquest exemple, el text s'emplenarà amb lletres comodí) → | e | d | e | l | e |
| | l | c | o | l | u |
| | m | n | e | s | x |

La lletra vermella en aquest cas seria la lletra comodí. Per últim, canviem les columnes de posició ordenant les lletres de la paraula clau per l'ordre alfabètic, tal i com s'indica a l'exemple:

| **A** | **E** | **M** | **S** | **T** |
|---|---|---|---|---|
| e | o | m | d | t |
| d | l | e | e | e |
| c | l | l | u | o |
| n | s | m | x | e |

Ara només caldria tornar a escriure el missatge per files, amb la qual cosa quedaria de la manera següent: "eomdtdleeeclluonsmxe". S'ha de tenir en compte que, en aquest mètode, les paraules clau no poden tenir lletres repetides o, si en tenen, s'han d'eliminar. Per desencriptar el missatge, només caldria invertir el procés utilitzat: coneixent la paraula clau, hauríem de classificar el missatge seguint l'ordre alfabètic de la paraula, i després canviar les columnes per tal d'obtenir la paraula clau en ordre (això significaria que el missatge també estaria en ordre).

- Xifratge de Cèsar: el primer mètode de xifrat d'un missatge a partir d'un algorisme de substitució el va utilitzar l'emperador romà Juli Cèsar. Va fer servir un sistema de substitució molt simple, en què s'havia d'intercanviar la lletra original del missatge per la que es trobava un nombre determinat de posicions més a la dreta en l'alfabet llatí. En el seu cas, s'intercanviava la lletra original per la que estava tres posicions a la dreta. Per tant, Juli Cèsar va crear un nou abecedari, el resultat del qual seria el següent.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |

La fila verda seria l'abecedari en el seu ordre habitual, i la fila taronja seria l'abecedari de Cèsar, que utilitzava per codificar missatges. Per xifrar un missatge, només caldria substituir la lletra original per la seva equivalent al segon abecedari. A més, no cal que aquest nou alfabet estigui desplaçat tres posicions a la dreta com a l'exemple, sinó que es pot fer amb qualsevol altre nombre.

Per exemple, voldrem xifrar el missatge "introducció a la criptografia". Primer de tot, haurem de decidir quantes posicions volem desplaçar l'abecedari i, en aquest cas, decidirem 5. Per tant, el nou abecedari (de taronja), quedaria així en comparació amb l'original (de verd):

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |

Ara, només hem de codificar el nostre missatge seguint el nou abecedari creat. El resultat seria el següent: NSYWTIHHNT F QF HWNUYTLWFKNF. Per desxifrar el missatge, només caldria invertir el procés seguit.

- Xifratge de Vigenère: al segle XVI, Vigenère va crear un mètode de xifrat que rep el seu nom i que és polialfabètic. Per explicar aquest xifratge, començaré posant un exemple. Suposem que volem codificar l'oració "m'agrada la criptografia". Primer de tot, necessitarem una paraula clau, que en aquest cas serà "matemàtiques". Ara, necessitarem crear un nou abecedari seguint la tècnica del xifratge de Cèsar, però haurem de crear-ne 26 diferents, un per a cada lletra. Un possible resultat seria el següent:

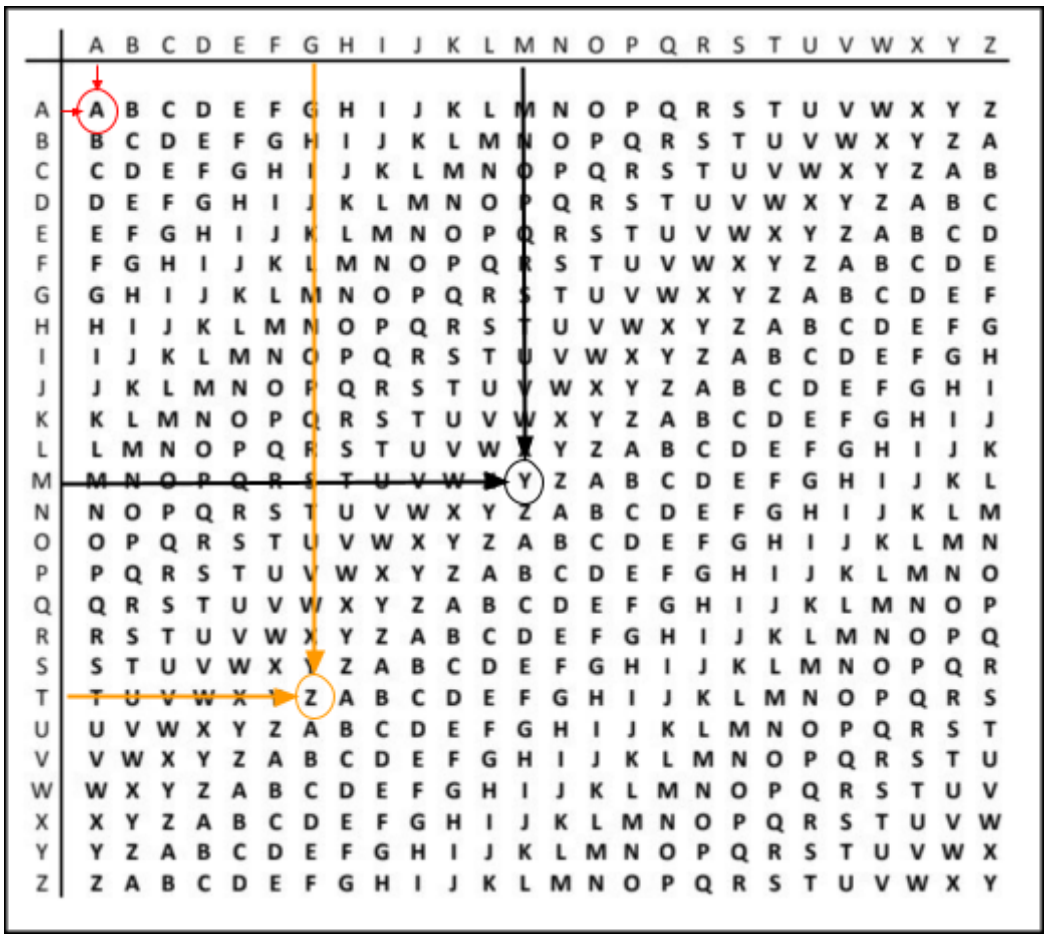
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | 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 | A |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Z | Z | 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 |

**Imatge 5.** Abecedari de Vigenère. *Imatge extreta de* https://moodle.feemcat.org/pluginfile.php/2202/mod_resource/content/1/CRIPTOGRAFIA-v1.pdf

Hem de col·locar el nostre missatge amb la paraula clau a sota, a cada lletra corresponent-li una de la paraula clau, de la manera següent:

M'AGRADA LA CRIPTOGRAFIA → missatge
M ATEMAT IQ UESMATE MATIQ → paraula clau

Sabent que la paraula clau és “matemàtiques”, veiem que comença per “m”. Llavors, hem de substituir la primera lletra del missatge (“m’agrada la criptografia”), que és la “m”, per la lletra que hi ha a la taula en la fila M (perquè és la primera lletra de la paraula clau) i la columna M (perquè és la primera lletra del missatge. Després hem de fixar-nos en la fila A, columna A; en la fila T, columna G…



**Imatge 6.** Xifratge de Vigenère. *Imatge extreta de* https://moodle.feemcat.org/pluginfile.php/2202/mod_resource/content/1/CRIPTOGRAFIA-v1.pdf

Així, obtenim el text xifrat següent: Y'AZVMDT TQ WVABTHKDAYQQ. Per desxifrar el missatge només caldria invertir el procés. El xifrat de Vigenère es va poder trencar, és a dir, va deixar de ser segur, quan els estudis es van centrar en descobrir la longitud de la clau. Malgrat tot, un dels inconvenients principals de tots els mètodes de xifrats amb clau explicats és com distribuir les claus entre l'emissor i el receptor sense que es puguin interceptar.

- El criptosistema de clau pública RSA: en un sistema criptogràfic de clau pública, la clau utilitzada per xifrar els missatges i la clau utilitzada per desxifrar-los són diferents. La part necessària per xifrar es dona a conèixer a tothom (és la clau pública), mentre que la part necessària per

desxifrar es manté secreta. D'aquesta forma, tothom pot xifrar missatges, però només la persona interessada pot desxifrar-los.

El mètode RSA és un dels mètodes de clau pública més emprats en l'actualitat. El seu nom prové de les inicials dels cognoms dels tres matemàtics americans que el van inventar: Ron Rivest, Ami Shamir i Leonard Adleman, l'any 1977.

Per poder xifrar i desxifrar missatges amb el mètode RSA és necessari tenir una clau RSA. Una clau RSA consta de tres nombres (N, e, d). Els dos primers són la part pública, i el tercer és la clau privada.

Els nombres que formen una clau RSA han de complir les següents condicions:

| | |
|---|---|
| $N = p \cdot q$ | el nombre N ha de descomposar-se com a producte de dos nombres primers senars diferents (p, q). |
| e | el nombre *e* ha de ser un nombre invertible mòdul $M = (p - 1) \cdot (q - 1)$ |
| d | el nombre *d* ha de ser l'invers del nombre *e* mòdul M. |

Alguns aclariments de la taula són:

1. L'operació mòdul consisteix en fer una divisió i obtenir-ne el residu. Per exemple: $542 \pmod{23} = 13$. Aquesta operació vol dir que, en dividir el nombre 542 entre 23, obtens un residu 13, i també es pot expressar de la manera següent: $542 = 13 \pmod{23}$.
2. Un nombre invertible d'un mòdul M és un nombre que pot tenir un invers.
3. Dos nombres són inversos si en multiplicar-los i fer el mòdul de M, el resultat és 1. Per exemple: $2 \cdot 3 = 1 \pmod{5}$. Aquesta operació vol dir que si dividim el nombre 6, que és igual a $2 \cdot 3$, entre 5,

obtindrem un residu d'1. Això significa que el 2 i el 3 són inversos per al mòdul 5.

Sabent això, ara comprovarem si la clau RSA següent és correcta, ja que mirarem si els seus termes (N, e, d) compleixen les característiques de la taula anterior. El nombre RSA serà el següent: (33, 3, 7).

1. Primer de tot, haurem de mirar si el 33 es pot descomposar com a producte de dos nombres primers senars diferents: $33 = 3 \cdot 11$. Sí que es compleix.
2. Després, haurem de mirar si *e* és un nombre invertible de mòdul $M = (p - 1) \cdot (q - 1)$.

$$M = (p - 1) \cdot (q - 1)$$
$$M = (3 - 1) \cdot (11 - 1) = 2 \cdot 10 = 20$$

$$e \cdot d = 1 \pmod{20} \rightarrow 3 \cdot 7 = 1 \bmod (20) \rightarrow 21 = 1 \bmod (20)$$

Efectivament, es comprova, ja que el seu invers és 7. Aquí també hem calculat l'últim pas, que era comprovar si *d* era invers de *e* mòdul M, i sí que ho és, ja que el resultat d'aquest mòdul és 1. Per tant, això significa que la clau RSA (33, 3, 7) és correcta.

Per xifrar un missatge amb el sistema RSA, només es necessita la part pública de la clau, és a dir, (N, e). Primer cal traduir el missatge a nombres. En aquest cas, utilitzarem la taula següent:

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

**Imatge 7.** Conversió de lletres a nombres. *Imatge extreta de* https://moodle.feemcat.org/pluginfile.php/2202/mod_resource/content/1/CRIPTOGRAFIA-v1.pdf

Un cop s'ha convertit cada lletra en un nombre, transformem aquest nombre elevant-lo a *e* i fent mòdul *N*.

Per exemple, si volem xifrar la paraula "Estalmat" utilitzant la clau pública (33, 3), haurem de seguir els passos següents:

1. Traduïm la paraula a una seqüència de nombres seguint la taula anterior: ESTALMAT = 05 19 20 01 12 13 01 20.
2. Ara calculem el nou valor de cada nombre seguint les instruccions anteriors: hem d'elevar-lo a *e* i fer el mòdul *N*.

$$5^3 \equiv 125 \equiv 26 \quad (\text{mòdul } 33)$$
$$19^3 \equiv 6859 \equiv 28 \quad (\text{mòdul } 33)$$
$$20^3 \equiv 8000 \equiv 14 \quad (\text{mòdul } 33)$$
$$1^3 \equiv 1 \equiv 01 \quad (\text{mòdul } 33)$$
$$12^3 \equiv 12 \quad (\text{mòdul } 33)$$
$$13^3 \equiv 19 \quad (\text{mòdul } 33)$$
$$1^3 \equiv 1 \equiv 01 \quad (\text{mòdul } 33)$$
$$20^3 \equiv 8000 \equiv 14 \quad (\text{mòdul } 33)$$

**Imatge 8.** Càlcul de la clau RSA. *Imatge extreta de* https://moodle.feemcat.org/pluginfile.php/2202/mod_resource/content/1/CRIPTOGRAFIA-v1.pdf

La paraula xifrada serà 2628140112190114. S'ha de tenir en compte que, en comptes de posar "1", hem posat "01", ja que és molt important respectar la longitud dels blocs.

Per desxifrar un missatge encriptat en el sistema RSA, tenim prou coneixent la clau privada *d* i el nombre *N*. Tan sols cal repetir el procés de xifrat que s'ha seguit, però calculant les potències amb l'exponent *d*, en comptes de *e*. Per exemple, per desencriptar el missatge que acabem de codificar, i com que la nostra clau RSA és (33, 3, 7), hauríem de fer el següent:

$26^7 \equiv 5 \pmod{33} \rightarrow 05 = E$
$28^7 \equiv 19 \pmod{33} \rightarrow 19 = S$
$14^7 \equiv 20 \pmod{33} \rightarrow 20 = T$
$1^7 \equiv 1 \pmod{33} \rightarrow 01 = A$
$12^7 \equiv 12 \pmod{33} \rightarrow 12 = L$
$19^7 \equiv 13 \pmod{33} \rightarrow 13 = M$
$1^7 \equiv 1 \pmod{33} \rightarrow 01 = A$
$14^7 \equiv 20 \pmod{33} \rightarrow 20 = T$

Els valors de les lletres s'obtenen a partir de la taula de sobre.

**Imatge 9.** Desxifrar la clau RSA. *Imatge extreta de* https://moodle.feemcat.org/pluginfile.php/2202/mod_resource/content/1/CRIPTOGRAFIA-v1.pdf

- Xifratge afí: aquest mètode de xifratge ens acostarà als actuals mètodes d'encriptació usats en la transmissió d'informació per internet. Va aparèixer l'any 1976, i va resoldre el problema explicat de com distribuir les claus entre l'emissor i el receptor mitjançant l'aritmètica modular. Per

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

**Imatge 10.** Conversió de lletres a nombres. *Imatge extreta de* https://moodle.feemcat.org/pluginfile.php/2202/mod_resource/content/1/CRIPTOGRAFIA-v1.pdf

explicar aquest mètode de xifratge, utilitzaré un exemple: suposem que els missatges només tenen lletres i espais en blanc (no hi ha accents, ni dièresis, ni comes, ni punts…). Assignarem a l'espai en blanc el valor de 0, i a les lletres, els nombres de la següent taula:

Treballarem en l'aritmètica del mòdul 27. Busquem un nombre *a* que tingui invers mòdul 27. Per a l'exemple de després agafarem *a* = 2, on veiem que es compleix que $2 \cdot 14 = 28 = 1 \pmod{27}$, cosa que significa que l'invers de 2 és 14, i *b* = 10.

Per exemple, suposem que volem enviar el missatge “adeu”:

1. Primerament, hem de traduir el missatge a una seqüència de nombres, d’acord amb l'assignació feta anteriorment a la taula.

   ADEU → (1, 4, 5, 21)

2. A continuació, hem de substituir cada nombre *x* d’aquesta seqüència pel nombre *y*, que ha de ser més gran o igual que zero i més petit o igual que 26, que compleixi:

   $a \cdot x + b = y \pmod{27}$ → aquest procés s’anomena transformació afí.

   En el nostre exemple, hauríem de fer les operacions següents:

   $1 \rightarrow 2 \cdot 1 + 10 = 12 \pmod{27}$

   $4 \rightarrow 2 \cdot 4 + 10 = 18 \pmod{27}$

   $5 \rightarrow 2 \cdot 5 + 10 = 20 \pmod{27}$

   $21 \rightarrow 2 \cdot 21 + 10 = 25 \pmod{27}$

3. Per últim, hem de traduir la nova seqüència de nombres (12, 18, 20, 25) a les lletres corresponents, que són “LRTY”, i enviar el missatge.

Per desxifrar el missatge, hem de recórrer el camí invers:

1. Hem de traduir les lletres rebudes a una seqüència de nombres:

   “LYTR” → (12, 18, 20, 25)

2. Hem de substituir cada nombre *x* d'aquesta seqüència pel nombre *y* que compleixi:

   (y - b) / a = x (mod 27)

   Per a poder fer-ho, cal poder dividir per *a* el mòdul 27, és a dir, que el nombre escollit *a* tingui invers mòdul 27. Prèviament, havíem escollit 2, que té 14 com a invers mòdul 27. Per tant, dividir per 2 mòdul 27 equival a multiplicar per 14 mòdul 27.

   $12 \rightarrow (12 - 10) \cdot 14 = 28 = 1 \pmod{27}$

   $18 \rightarrow (18 - 10) \cdot 14 = 112 = 4 \pmod{27}$

   $20 \rightarrow (20 - 10) \cdot 14 = 140 = 5 \pmod{27}$

   $25 \rightarrow (25 - 10) \cdot 14 = 210 = 21 \pmod{27}$

3. Finalment, hem de traduir la seqüència numèrica resultant a lletres:

   (1, 4, 5, 21) → ADEU

### 3.4. USOS I IMPLEMENTACIONS DE LA CRIPTOGRAFIA

Des dels seus orígens, la criptografia s'ha emprat per xifrar i desxifrar missatges per tal que emissor i receptor l'entenguessin sense que la resta ho fessin. Per tant, al llarg de la història la seva aplicació bàsica ha estat l'encriptació de missatges.

Actualment, però, el camp de la criptografia és molt més ampli. Es pot encabir l'aplicació actual de la criptografia en els següents àmbits:

### 3.4.1. Seguretat de les comunicacions

És la principal aplicació de la criptografia. Totes les xarxes d'ordinadors les utilitzen, ja que permeten establir canals segurs sobre xarxes que no ho son. A més a més, amb la potència de càlcul actual i utilitzant algoritmes de xifrat s'aconsegueix la privacitat sense perdre la velocitat de transferència.

### 3.4.2. Identificació i autentificació

Gràcies a l'ús de les signatures digitals és possible identificar a un individu o validar l'accés a un recurs en un entorn de xarxa amb més garanties que amb els sistemes d'usuari i clau tradicionals.

En la signatura digital s'han de tenir en compte tres paràmetres: la confidencialitat, l'autenticació i la integritat. Per aconseguir-ho es creen dues claus, la clau privada, que només coneix l'usuari i s'utilitza per generar la signatura i la clau pública, que serveix per desxifrar-la. Són claus complementàries, i només la pública pot desxifrar el que ha xifrat la privada i a l'inrevés. És impossible esbrinar la clau privada a través de la pública.

### 3.4.3. Certificació

La certificació és un esquema mitjançant el qual agents fiables, com una entitat certificadora, valida la identitat d'agents desconeguts, com usuaris reals, com a representants legals d'entitats, com per exemple empreses. D'aquesta manera, particulars es converteixen en representants legals de les empreses amb aquest certificat digital.

### 3.4.3. Comerç electrònic

Mitjançant la utilització de canals segurs i de mecanismes d'identificació es possibilita el comerç electrònic. D'aquesta manera, realitzant transaccions en xarxa amb targes o amb plataformes digitals, es possibilita el comerç electrònic, ja que tant les empreses com els usuaris tenen garanties de que les operacions no poden ser espiades, reduint-se el risc de robatoris o frau.

# 4. LA MÀQUINA ENIGMA

## 4.1. HISTÒRIA DE LA MÀQUINA ENIGMA

La màquina Enigma va ser una màquina utilitzada per Alemanya durant la segona guerra mundial per tal d'enviar missatges encriptats. Va causar bastants maldecaps als aliats, sobretot a la zona de l'Atlàntic Nord, on els combois de material procedents dels Estats Units queien pressa dels submarins alemanys que es comunicaven entre sí utilitzant el codi que generava aquesta màquina.

La va inventar un enginyer alemany, Arthur Scherbius, que era expert en electromecànica. Després de la Primera Guerra Mundial va aplicar la tecnologia que existia per millorar els sistemes criptogràfics de l'exèrcit, i va patentar la seva idea al 1918. Consistia en aplicar un xifrat de Vigenère, és a dir, un algoritme de substitució d'unes lletres per unes altres. Scherbius es va associar amb Willie Korn, que tenia una empresa anomenada Enigma Chiffiermanschinen AG, i amb els diners d'aquest, van millorar el disseny. Al 1923 van presentar la màquina a l'Exhibició Postal Internacional de Berlín, i servia per xifrar els secrets comercials.

**Imatge 11.** Màquina Enigma. *Imatge extreta de https://www.abc.es/cultura/abci-enigma-maquina-cambio-rumbo-guerra-mundial-202006030044_noticia.html*

Al 1933 Alemanya va nacionalitzar l'empresa Enigma Chiffiermaschinen AG i va passar a equipar tot l'exèrcit alemany, que va utilitzar aquestes màquines de xifrat. A més se'ls va afegir un quart cilindre per complicar més el fet de desxifrar els missatges.

Durant la Segona Guerra Mundial, Alemanya comptava amb un gran avantatge, atès que el codi d'Enigma era pràcticament indesxifrable. A més a més, l'exèrcit alemany canviava cada dia el codi a utilitzar, de forma que els Aliats només disposaven d'un dia per desxifrar-lo.

Donada la seva gran complexitat, és difícil de creure com es va poder desxifrar. Però va ser possible per diverses causes. La primera va ser per la

comercialització del principi de funcionament de la màquina a l'exposició del 1923, i tot i que la versió militar era més complexa, el principi de funcionament ja es coneixia. La segona va ser el fet que la codificació de missatges obligava als operadors a introduir tres lletres, dos cops, en iniciar el missatge, un codi per reconèixer-ho. Aquesta seqüència no es modificava, i, per tant, aquest patró sempre es repetia. Aquest fet el va aprofitar Marian Rejewski per tal de desxifrar el codi gràcies a les seves màquines anomenades bombes criptològiques, que no eren més que màquines Enigma de processament en paral·lel que buscaven les codificacions possibles. I la tercera va ser la captura del submarí alemany U-110 el 9 de maig del 1941 per part de la Royal Navy, de forma que es va fer amb una màquina enigma i el seu llibre de claus. Es va fer creure a la opinió pública que el submarí havia estat enfonsat per tal que els alemanys no canviessin les claus.

A partir d'aquí l'exèrcit alemany va començar a tenir moltes pèrdues. Van evolucionar Enigma i van crear una nova màquina, la M4, que va ser vençuda per Colossus, un computador dissenyat per desxifrar els codis alemanys.

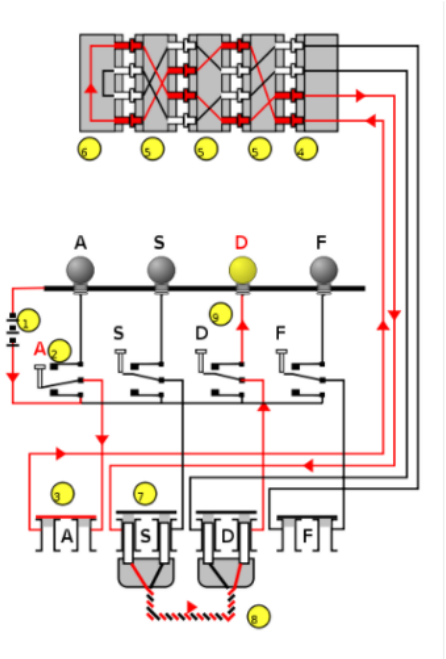
### 4.2. FUNCIONAMENT DE LA MÀQUINA ENIGMA

La màquina Enigma era un dispositiu electromecànic, és a dir, tenia una part elèctrica i una mecànica. El mecanisme consistia en unes tecles amb les lletres de l'alfabet, com una màquina d'escriure, però que en comptes d'accionar la palanca amb la lletra sobre una banda tintada, era un interruptor que accionava dispositius elèctrics que feien moure uns cilindres rotatoris. L'usuari de la màquina Enigma havia d'escriure el seu missatge i anotar les lletres que la màquina li retornava (a través d'un alfabet que s'anava il·luminant). El codi a utilitzar es fixava amb les posicions dels cilindres que constaven, cadascú, de vint-i-sis cables que es connectaven al teclat, però, amb la particularitat que el primer cilindre girava una vint-i-sisena volta després de cada pulsació, de manera que la posició de les connexions anava canviant amb cada entrada del teclat, obtenint un xifrat polialfabètic. A més a més, per donar major complexitat, el segon cilindre només feia un gir quan el primer havia completat els vint-i-sis girs, i el tercer quan el segon havia fet, al seu torn, els vint-i-sis girs. També va afegir la possibilitat que els rodets poguessin intercanviar la

posició de manera que el nombre d'alfabets que es podien arribar a aconseguir eren de 105.456.

Per acabar de complicar-ho més, el sistema comptava amb sis cables de connexió que també permetien introduir modificacions, atès que es podien connectar en vint-i-sis llocs, el que produïa un total de 100.391.500 formes diferents de connectar els cables, que units als 105.456 alfabets que s'aconseguien feia un total de 3.283.883.513.796.974.198.700.882.069.882.752.878.379.955.261.095.623.685.444.055.315.226.006.433.616.627.409.666.933.182.371.154.802.769.920.000.000.000 possibilitats diferents de codificació.

A continuació exposem un exemple pràctic de codificació amb la màquina Enigma. La figura mostra un esquema de la configuració del circuit de la màquina enigma, només amb quatre lletres.




**Imatge 12.** Esquema de codificació amb Enigma. *Imatge extreta de https://histinf.blogs.upv.es/2011/11/04/2248/*

Els diferents nombres són el següent:

1. Flux de corrent de la bateria
2. Interruptor bidireccional de la tecla «A» (pulsat).
3. Sòcol tancat de «A».

4. Entrada a les rodes dentades.

5. Rodes dentades

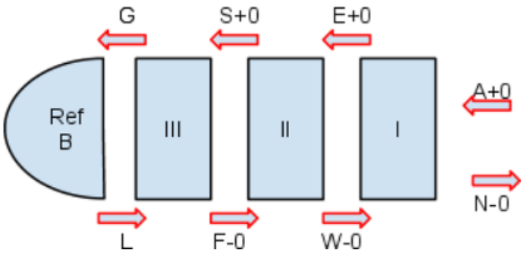
6. Reflector (torna el corrent per un altre camí).

7. Sòcol obert de «S».

8. Cable d'intercanvi de lletres (de «S» a «D»)

A l'exemple es veu com es polsa la tecla «A» les rodes tornen una «S», però com estan intercanviades la «S» i la «D», es produeix una «D».

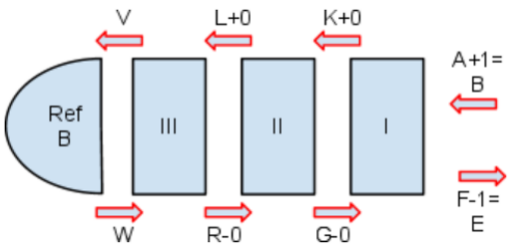
Com a exemple pràctic, suposem que es vol enviar un missatge codificat amb Enigma, començant per la lletra A i utilitzant les tres rodes, I, II i III de dreta a esquerra. Per fer-ho més simple, s'utilitzarà com a posició inicial 0 per a les tres rodes.




**Imatge 13.** Codificació amb Enigma (I). *Imatge extreta de https://histinf.blogs.upv.es/2011/11/04/2248/*

Començant des de la dreta, es veu que per a desplaçament 0, «A» es transforma en «E». A la segona roda «E» es transforma en «S», i a la tercera «S» es transforma en «G». Finalment el reflector conferteix «G» en «L». Tornem d'esquerra a dreta i «L» es converteix en «F», després «F» en «W», i finalment «W» en «N». Aquest seria el missatge xifrat amb la clau 000.

Un cop acabat aquest xifrat, la màquina passaria automàticament a la posició 001, de forma que si es torna a introduir una «A» es convertiria en una «E».



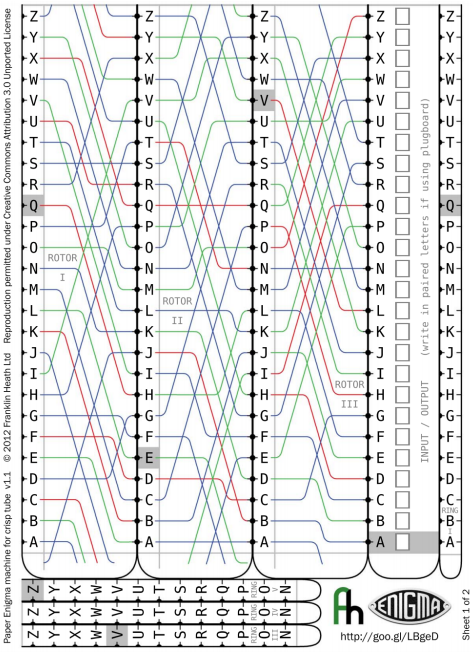

**Imatge 14.** Codificació amb Enigma (II). *Imatge extreta de https://histinf.blogs.upv.es/2011/11/04/2248/*

# 5. DISSENY I CREACIÓ DE SIMULACIONS DE LA MÀQUINA ENIGMA

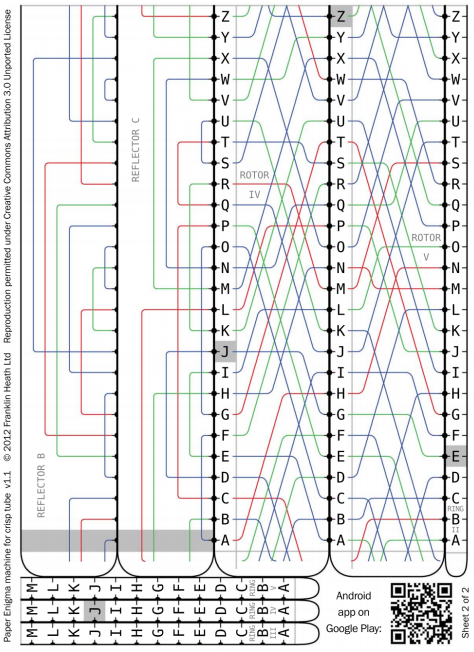
## 5.1. MÀQUINA ENIGMA MANUAL

Per tal d'obtenir una eina de codificació i descodificació senzilla de crear i eficaç, no cal fer cap programa informàtic. Els materials i objectes necessaris per crear una màquina Enigma són els següents:

- Les plantilles de la següent imatge dels rotors, reflectors i lletres d'entrada i sortida.




**Imatge 15.** Paper 1 per imprimir. *Imatge extreta de https://fhcouk.files.wordpress.com/2012/05/pringlesenigma3a4.pdf*

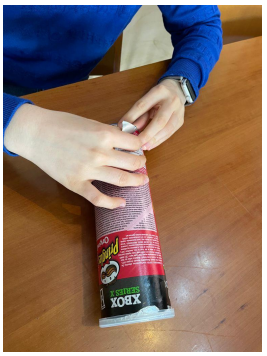



**Imatge 16.** Paper 2 per imprimir. *Imatge extreta de https://fhcouk.files.wordpress.com/2012/05/pringlesenigma3a4.pdf*

- Un pot de patates Pringles de mida llarga, és a dir, de 23 centímetres de longitud.
- Unes tisores.
- Un tub de cola o barra d'adhesiu.
- Un regle.
- Un bolígraf o retolador.

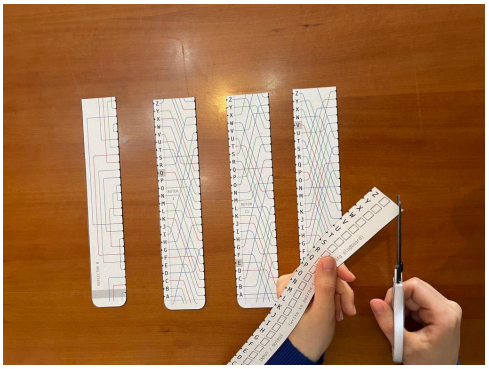
Els passos a seguir per tal d'obtenir una màquina Enigma manual són els següents:

1. Imprimir les plantilles amb la mesura real de l'impressora, és a dir, sense marges. Això és molt important, ja que, si no, després la mida dels papers no serà suficientment gran per al pot de Pringles.
2. Folrem el pot de patates amb paper adhesiu d'alumini. Aquest pas és opcional, però estèticament és recomenable.

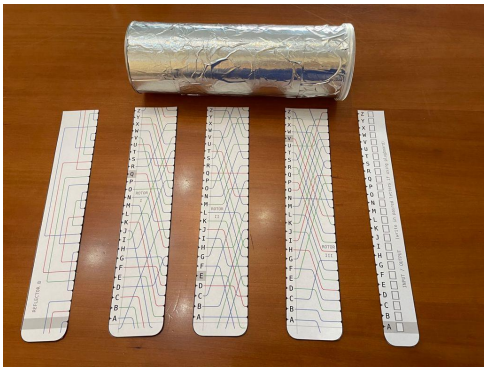


**Imatge 17.** És recomanable folrar el pot. *Font pròpia.*

3. Del paper 1 de les plantilles impresses, hem de retallar la tira *input / output*, i les tires dels rotors I, II i III. Del paper 2, hem de retallar el reflector B.



**Imatge 18.** Tires retallades. *Font pròpia.*



**Imatge 19.** Tires retallades i pot folrat. *Font pròpia.*

4. Agafem el regle i el bolígraf o retolador i fem una línia al pot de Pringles tal i com s'indica a la imatge.

**Imatge 20.** Línia al pot de Pringles. *Font pròpia.*

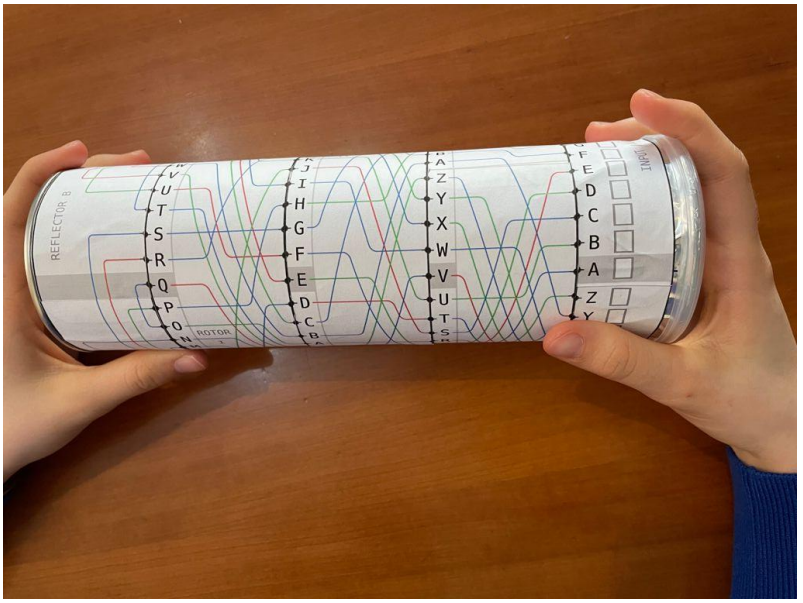
5. Posem cola a la base del tub de Pringles i hi enganxem la tira de paper del reflector B, però la línia que hem dibuixat prèviament ha de coincidir amb amb el centre de la part grisa del nostre reflector. La tira de paper del reflector B ha de quedar fixa, completament enganxada al pot.

**Imatge 21.** S'ha d'enganxar el reflector. *Font própia.*

**Imatge 22.** Reflector enganxat. *Font própia.*

6. Al costat del reflector B hem de posar el rotor 1, però en aquest cas només hem de posar cola als extrems de la tira de paper, per tal que aquesta pugui rotar sobre el tub.

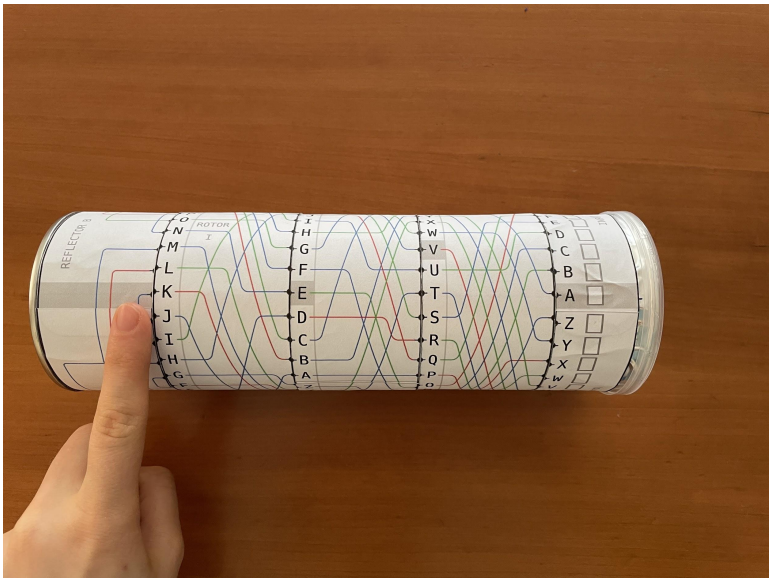
7. Hem de fer el mateix amb els rotors II i II. És important que les tires estiguin col·locades el més juntes possible.
8. Finalment, hem d'enganxar la tira *input / output* de la mateixa manera en la qual vam enganxar el reflector B per tal que sigui fixa, i hem de tenir en compte que la línia que havíem dibuixat ha de coincidir amb el centre de la franja gris d'aquesta tira.




**Imatge 23.** Màquina Enigma manual finalitzada. *Font pròpia.*

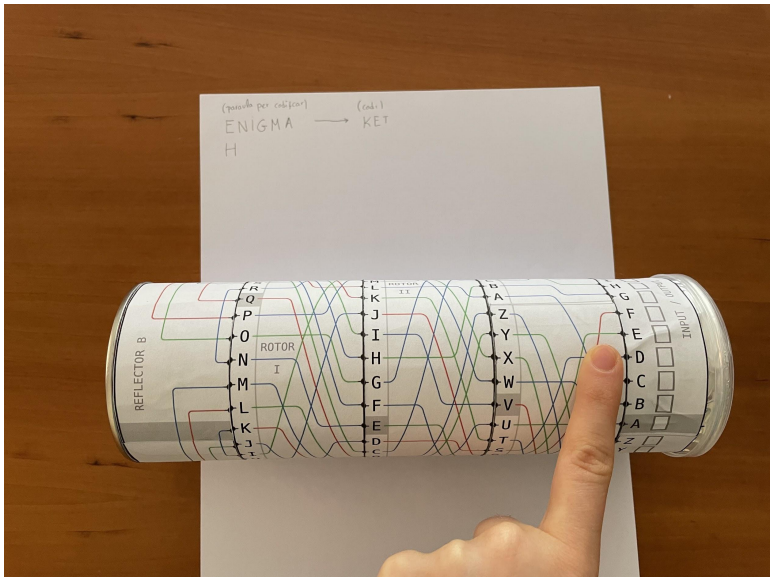
Per codificar amb aquesta màquina enigma manual s'ha de fer el següent:

1. Hem d'escollir primerament el codi inicial que volem. Per exemple, si vull que el meu codi inicial sigui KET, hauré de girar els tres rotors, per tal que les lletres K (rotor I), E (rotor II) i T (rotor III) quedin alineades amb la franja grisa del reflector i de la tira *input / output*. Per exemple, voldrem codificar la paraula "enigma".



**Imatge 24.** Introduïm el nostre codi a la màquina Enigma. *Font pròpia.*

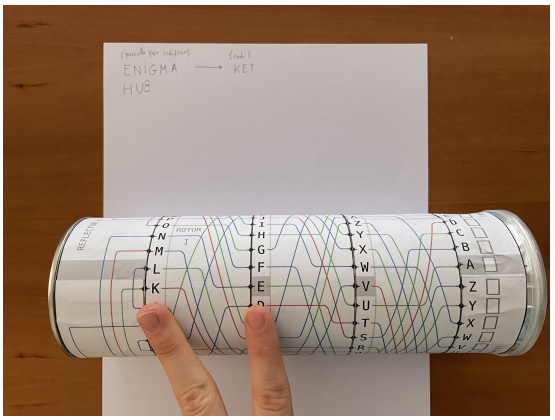
2. Hem de girar una vegada el rotor III. Després, hem de buscar la primera lletra de la nostra paraula codificada a *input / output*, i seguir tot el camí de línies passant pels rotors, arribant al reflector i tornant a passar pels rotors, fins a arribar a una lletra de la tira *input / output*, que serà diferent a la original. Així, haurem codificat la nostra primera lletra.




**Imatge 25.** Girem una vegada el rotor II i codifiquem. *Font pròpia.*

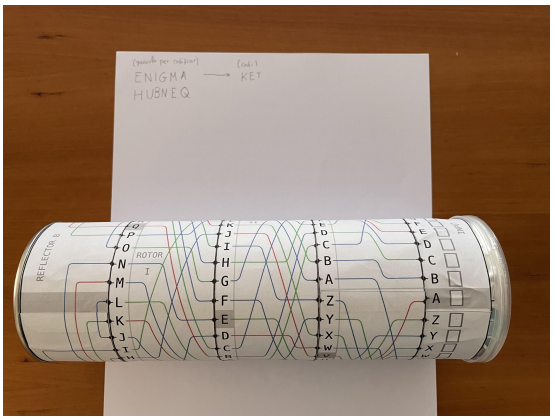
3. Haurem de fer això amb totes les lletres, però hem de tenir en compte el següent: si en codificar una lletra, un dels rotors té la franja gris que coincideix amb la part grisa del reflector i de la tira *input / output*, hem de

girar, no només aquest rotor, si no també el següent. Per exemple, en codificar la paraula “enigma” amb la clau KET, passa això. Quan arribem a la lletra “i”, hem de girar el rotor III, però com que hem girat la lletra “v”, que és de color gris, també hem de girar el següent rotor (el número II). Malgrat això, ens trobem que al rotor dos tenim la lletra “e”, que també és grisa, amb la qual cosa també haurem de girar una vegada el rotor I.




**Imatge 26.** Han coincidit franges grises als rotors II i III, així que hem de girar els tres rotors. *Font pròpia.*

4. Haurem de procedir d’aquesta manera fins al final, i ja tindrem la nostra paraula codificada. En el meu cas, la paraula “enigma” amb la clau KET es codificaria com “hubneq”.




**Imatge 27.** Ja hem codificat el nostre missatge. *Font pròpia.*

## 5.2. MÀQUINA ENIGMA PROGRAMADA AMB SCRATCH

### 5.2.1. SCRATCH

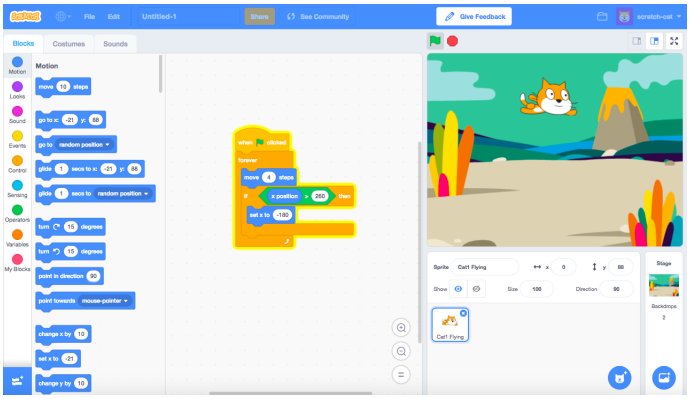
Scratch és un llenguatge de programació de software lliure que va crear el MIT (també conegut com Massachusetts Institute of Technology) i que serveix per programar històries interactives, jocs o animacions, entre d'altres. A més, permet difondre fàcilment els programes propis amb la resta de membres de la comunitat Scratch. Serveix per a iniciar-se al món de la programació, i tothom pot aprendre a utilitzar-lo. El seu nom prové de la paraula anglesa *scratching*, que, en llenguatges de programació, representa els trossos de codi que es poden reutilitzar, combinar i adaptar per a nous usos.

A l'entorn d'Scratch podem trobar principalment dos tipus d'elements: els objectes (o *sprites* a la nomenclatura d'Scratch) i les accions o blocs de comportament, que podrem combinar per aconseguir que els objectes actuïn d'una manera determinada. Les accions que programarem amb Scratch tenen forma de codi en blocs, cosa que fa que sigui un llenguatge de programació força intuïtiu i ideal per a programadors principiants.

Les accions que es poden dur a terme a Scratch estan dividides en els següents grups, encara que poden haver-hi més:

- Moviment: serveix per moure o girar un objecte per la pantalla.
- Aparença: canvia la visualització o característiques dels personatges i del fons.
- So: fa sonar diferents seqüències d'àudio.
- Events: inicien diferents accions en bloc depenent de les circumstàncies.
- Control: hi ha estructures condicionals.
- Sensors: els personatges poden interaccionar amb el seu entorn o elements creats per l'usuari.
- Operadors: hi ha operadors matemàtics i generadors aleatoris de nombres, entre d'altres.
- Variables: hi ha blocs que emmagatzemen informació i et permeten actuar amb ella.
- Els meus blocs: pots crear nous blocs personalitzats.

- Llapis: permet dibuixar controlant diferents aspectes, com la mida del pinzell, el color de la pintura o l'ombra.

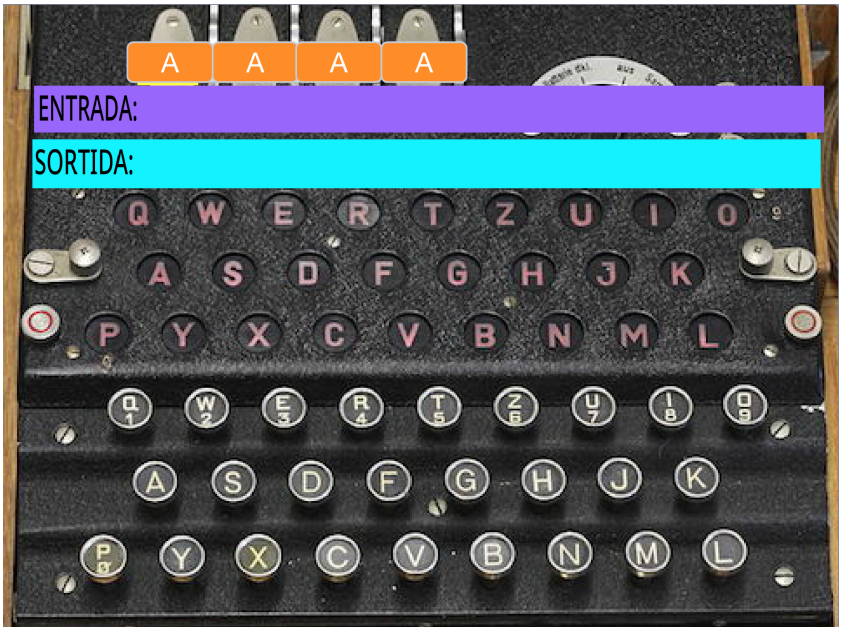


**Imatge 28.** Programació amb Scratch. *Imatge extreta de https://ligadegenios.com/producto/curso-de-programacion-con-scratch/*

### 5.2.2. PROGRAMACIÓ DE LA MÀQUINA ENIGMA

Per a programar la meva màquina Enigma he utilitzat el programa Scratch, ja que aquest llenguatge de programació ja és conegut per mi i que he utilitzat prèviament. A més, amb la programació de la màquina Enigma, podré aprendre a programar a nivell avançat amb Scratch.

La pantalla del programa que he realitzat és la següent:




**Imatge 29.** Escenari del meu programa de la màquina Enigma. *Font pròpia*

A aquest fons es poden distingir principalment quatre elements: una imatge del teclat de la màquina Enigma, els rotors (que, en aquest cas, estan posats en la combinació AAAA), una caixa d'informació on es mostraran les lletres introduides i una caixa de sortida, on es mostrarà el missatge codificat.

Exposaré el codi creat en la programació explicat per personatges. Al meu programa he creat cinc personatges:

- El text o lletres d'entrada (que aniran a la caixa lila d'entrada explicada a l'escenari).
- El text o lletres de sortida (que aniran a la caixa blava de sortida explicada també prèviament).
- Una caixa de color groc que permetrà canviar manualment el codi inicial de la màquina (és a dir, permetrà canviar la lletra dels rotors).
- Una llum de color gris que il·luminarà al teclat de la màquina Enigma la lletra que estem introduint amb el teclat de l'ordinador.
- Una llum de color blanc que il·luminarà al teclat de la màquina la lletra de sortida que es generarà.

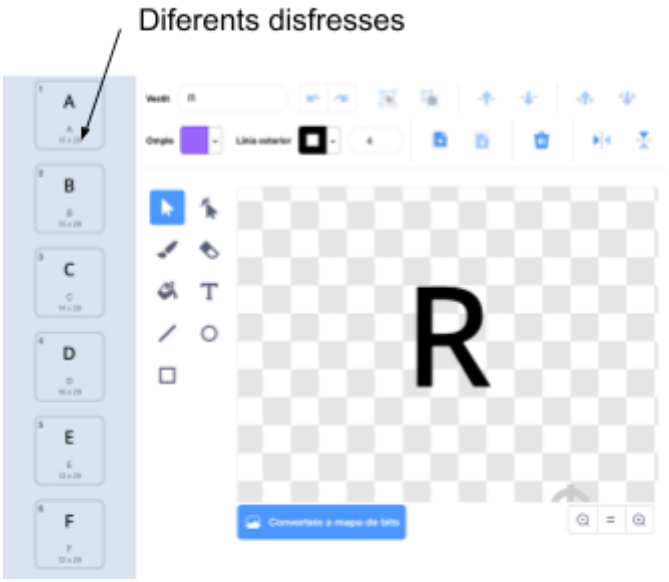

**Imatge 30.** Personatges del programa. *Font pròpia*

A continuació explicaré els diferents codis creats per als personatges i per a l'escenari, encara que estaran relacionats entre si per aconseguir el funcionament òptim de la simulació de la màquina Enigma.

**Lletres d'entrada**

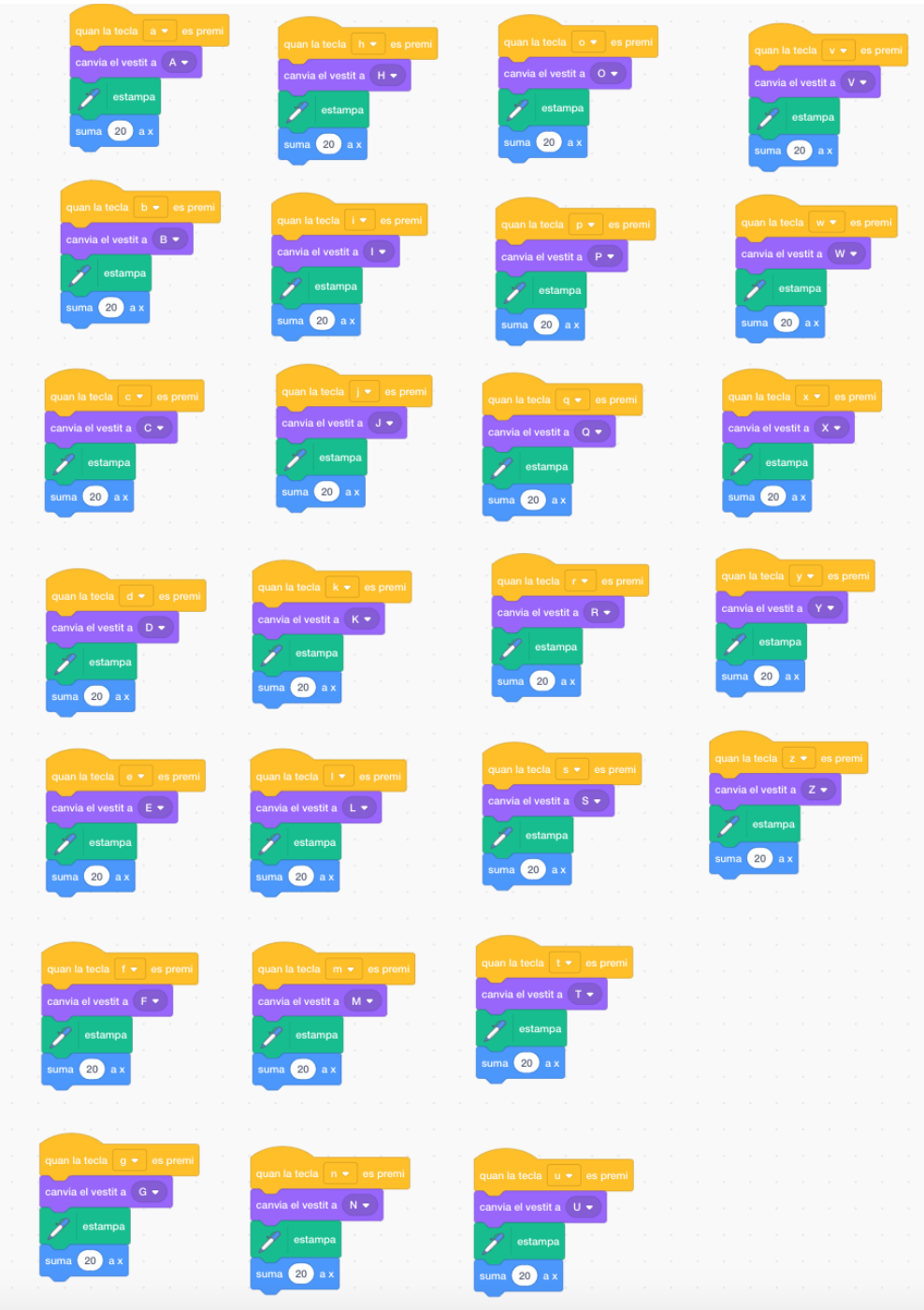

**Imatge 31.** Codi del text d'entrada. *Font pròpia*

El significat d'aquest codi és el següent: quan s'iniciï la simulació, és a dir, quan es premi la bandera verda, tot el text previ que s'ha escrit en altres simulacions s'esborrarà. Després, aquest personatge s'amagarà per a què no es vegi cap lletra quan encara l'usuari no ha premut cap lletra del teclat. Finalment, el personatge anirà a una posició determinada, que se situa al rectangle lila de l'escenari, just després de la paraula "entrada".




**Imatge 32.** Disfresses del text d'entrada. *Font pròpia*

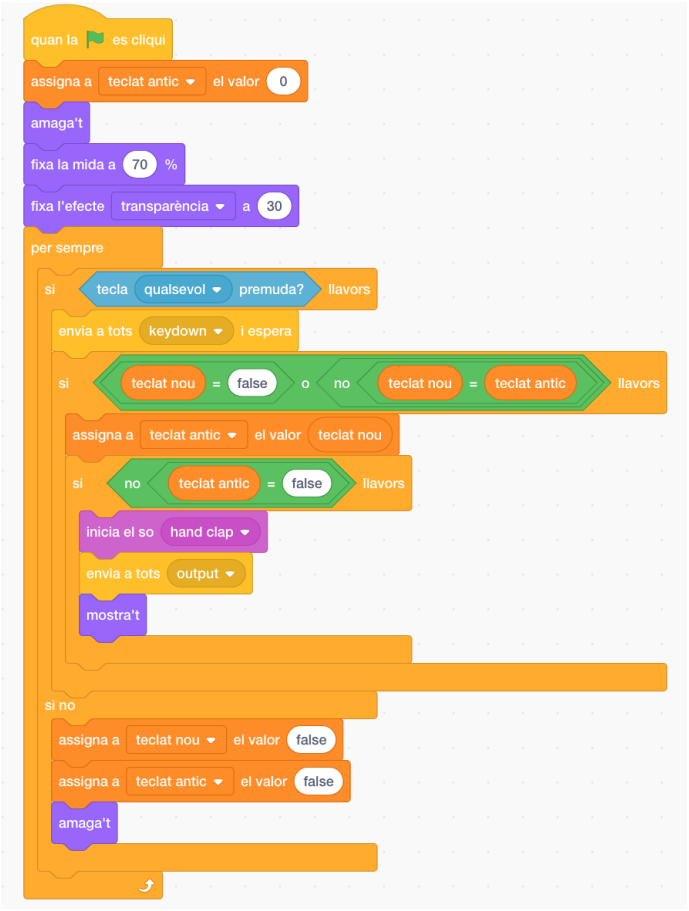
Com que aquest personatge és una única lletra, he creat diferents disfresses. Cada una serà una lletra diferent de l'abecedari, i quan l'usuari premi una tecla, aquest personatge canviarà la seva disfressa per ser la lletra corresponent.




**Imatge 33.** Codi del text d'entrada. *Font pròpia*

Aquests blocs de codi signifiquen el següent: quan l'usuari premi al teclat la lletra A, per exemple, el personatge canviarà la seva disfressa per la lletra A, s'escriurà o marcarà a l'espai al qual està (si no fessim aquesta ordre, el personatge es mouria per l'espai, però les diferents lletres que s'han premut no es quedarien a la pantalla, desapareixerien quan es premés una altra tecla), i es mourà 20 unitats en l'eix X. Aquesta última instrucció és necessària, perquè, així, la següent lletra estarà al costat de la escrita prèviament, i aconseguirem una línia de text seguida.

**Llum que il·lumina les lletres d'entrada**

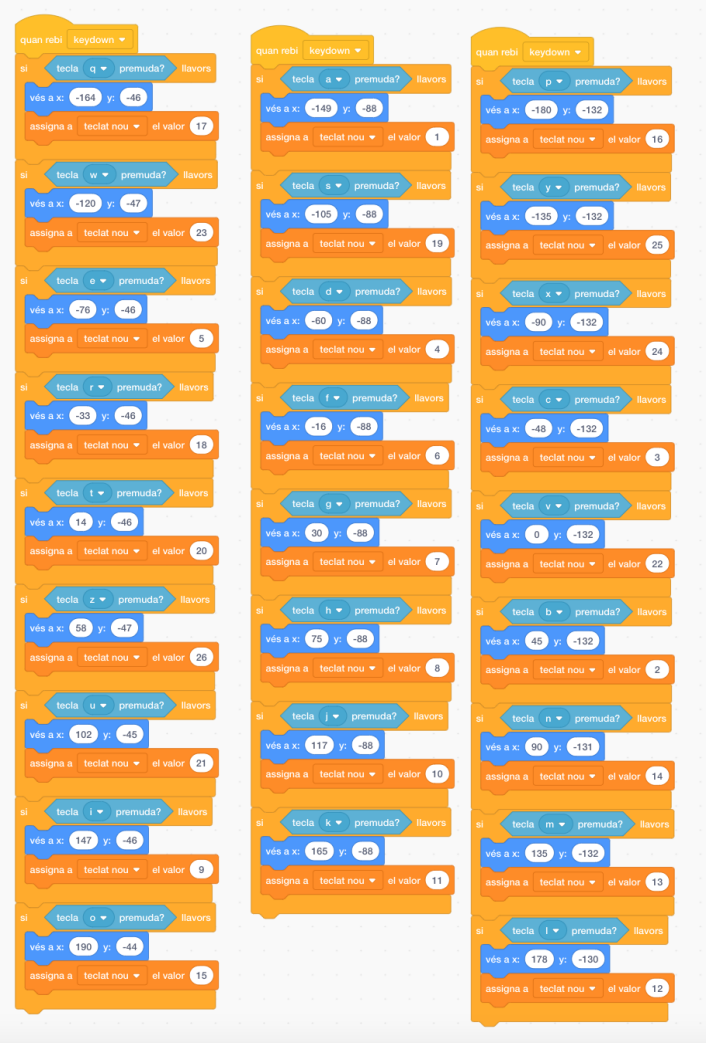



**Imatge 34.** Codi de la llum que il·lumina el text d'entrada. *Font pròpia*

Aquest codi començarà a executar-se quan la bandera verda es premi. Primer de tot, haurem creat una variable, anomenada *teclat antic*, i li donarem el valor de 0. Després, amagarem el nostre personatge, que és un cercle de color gris, perquè quan no hi ha cap tecla premuda, no volem que l'usuari el vegi. A continuació, com que el personatge és massa gran per a la mesura d'una tecla de l'escenari, haurem de reduir la seva mida, i en comptes d'estar al 100%, estarà al 70%. A més, com que en un primer moment el nostre cercle és completament opac, haurem de fixar la seva transparència al valor 30, en comptes de a 0.

Ara, haurem d'establir un bucle repetitiu, és a dir, crearem una sèrie d'instruccions que farem que s'executin per sempre. Si qualsevol tecla es prem, enviarem un missatge (anomenat *keydown*) i esperarem. Posteriorment,

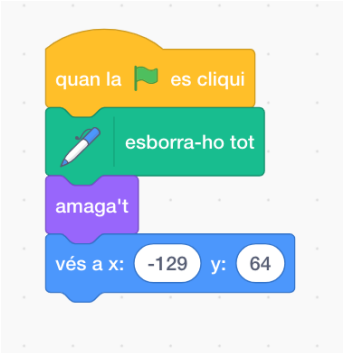
explicaré on utilitzarem aquest missatge, però el que farà serà veure si hi ha alguna tecla de l'abecedari premuda, i assignar-li un valor numèric. Per exemple, si la lletra A està premuda, a la variable *teclat nou* li atorgarem el valor 1, i així successivament. Si el valor de *teclat nou* és fals o el seu valor no és igual a la variable *teclat antic* (a la qual li havíem donat el valor 0 prèviament), iniciarem una sèrie d'accions: li donarem a la variable de *teclat antic* el mateix valor que a *teclat nou*. Si el valor de *teclat antic* és diferent a fals, iniciarem un so, que imitarà el que es produeix quan premem una tecla d'una màquina d'escriure, per exemple, enviarem un missatge anomenat *output* i farem que el personatge es mostri. Si no hi ha premuda cap tecla, *teclat nou* i *teclat antic* tindran el valor de fals, i el personatge s'amagarà.




**Imatge 35.** Codi de la llum que il·lumina el text d'entrada. *Font pròpia*

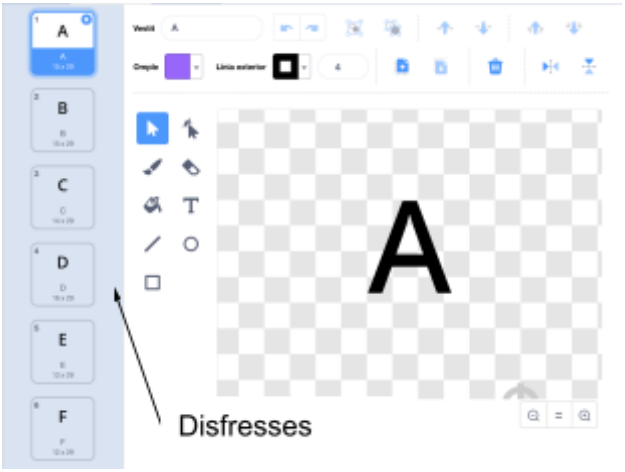
Aquest codi, que també és de la llum que il·lumina el text d'entrada, s'iniciarà quan rebi el missatge *keydown*. Aquí, podem apreciar diverses estructures condicionals. Per exemple, si la lletra Q està premuda, llavors farem que el nostre personatge vagi a les coordenades de l'escenari on es troba el dibuix de la lletra Q, i la variable *teclat nou* rebrà el valor numèric de 17, perquè a l'abecedari, aquesta lletra es correspon al número 17. Farem això amb totes les altres lletres, però, evidentment, canviarem les coordenades i el valor que li donarem a la variable *teclat nou* (per exemple, en el cas de la A, *teclat nou* serà igual a 1).

### Lletres de sortida



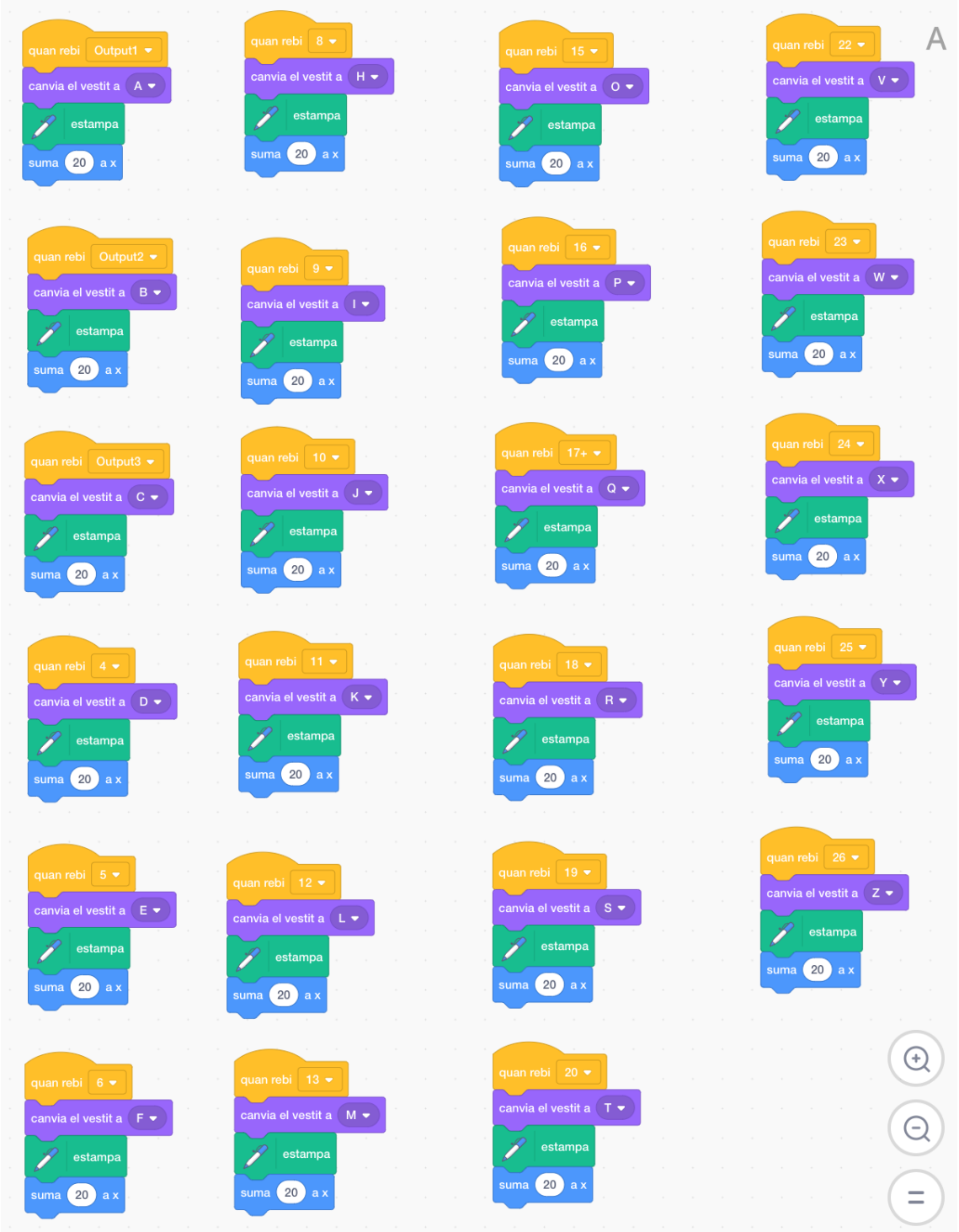

**Imatge 36.** Codi del text de sortida. *Font pròpia*

Aquest codi comença quan es prem la bandera verda i, per tant, s'inicia la simulació. S'esborraran tots els textos d'altres simulacions prèvies i, a continuació, el personatge s'amagarà, perquè així l'usuari no veurà cap lletra si no ha tocat el teclat. Finalment, farà que aquest personatge vagi a unes coordenades determinades, just després de la paraula "sortida" al rectangle blau.




**Imatge 37.** Disfresses del text de sortida. *Font pròpia*

A aquesta imatge podem veure les diferents disfresses que té. Tal i com passa al text d'entrada, aquest personatge només és una única lletra, però cada disfressa és una lletra diferent de l'abecedari. Quan es codifiqui el text d'entrada i s'escrigui el de sortida, el personatge canviarà el seu vestit per tal de ser la lletra corresponent.




**Imatge 38.** Codi del text de sortida. *Font pròpia*

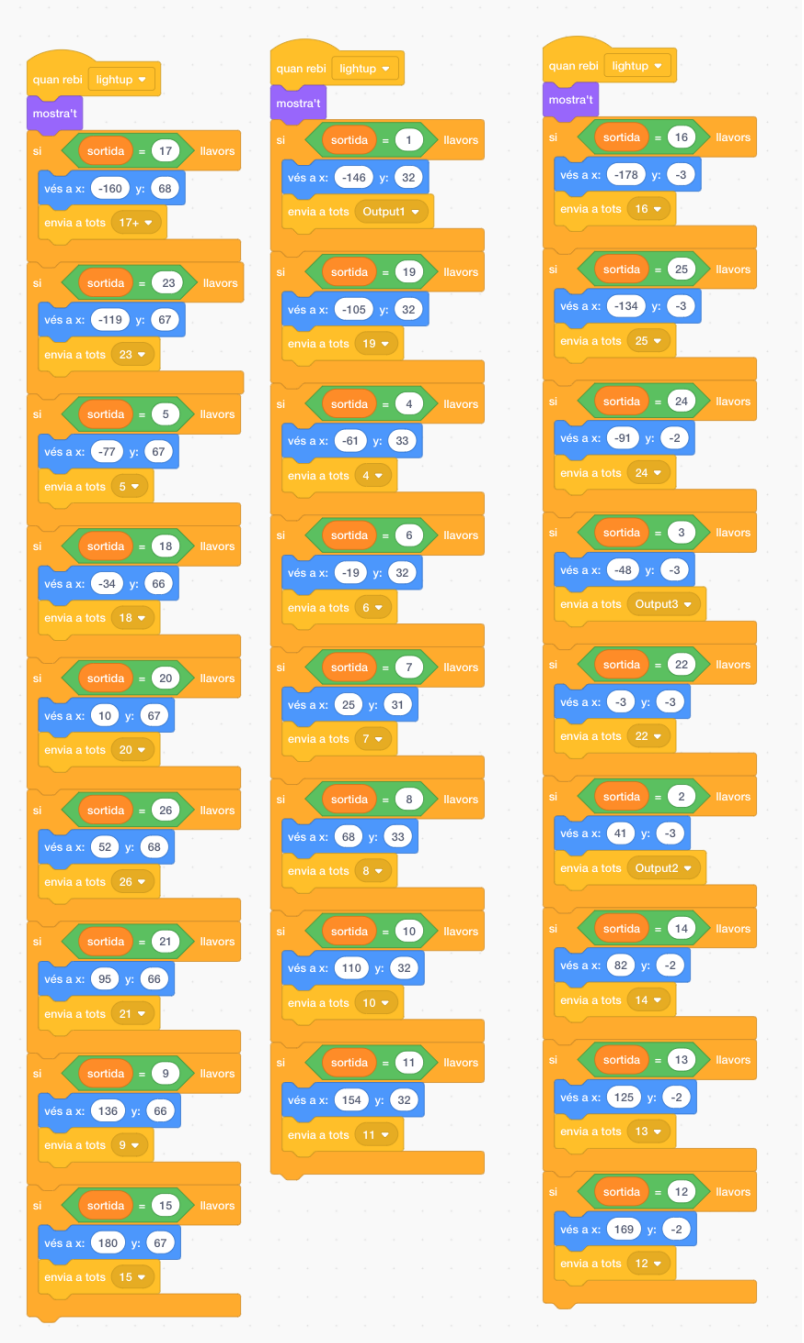
El significat d'aquestes accions és el següent: per exemple, quan s'enviï el missatge *output1* i aquest personatge el rebi, significarà que la lletra de sortida, és a dir, la lletra codificada és la número 1 o la A. Per això, haurà de canviar la seva disfressa a la de la lletra A, estampar-la o escriure-la i moure's 20 unitats cap a la dreta, amb la qual cosa la següent lletra s'escriurà al costat. Aquest

codi serà igual per a totes les lletres, però canviarà el missatge rebut (per exemple, amb la lletra B, el missatge serà *output2*) i la disfressa escollida.

**Llum que il·lumina les lletres de sortida**


**Imatge 39.** Codi de la llum que il·lumina el text de sortida. *Font pròpia*

Quan s'iniciï el programa, és a dir, quan l'usuari premi la bandera verda, encara no començaran a executar-se accions de moviment d'aquest personatge, però sí que establirem certs paràmetres del seu aspecte. Aquest personatge serà un cercle blanc, però la seva mida no serà l'adequada, així que, en comptes de deixar-la al 100%, la posarem al 60%. A més, tampoc volem que sigui completament opac, així que fixarem l'efecte de transparència al valor de 60 en comptes de 0. També augmentarem la seva brillantor, i la fixarem a 50 unitats. Per últim, farem que el personatge s'amagui, ja que no haurà d'aparèixer fins que l'usuari premi una tecla i aquesta es codifiqui.

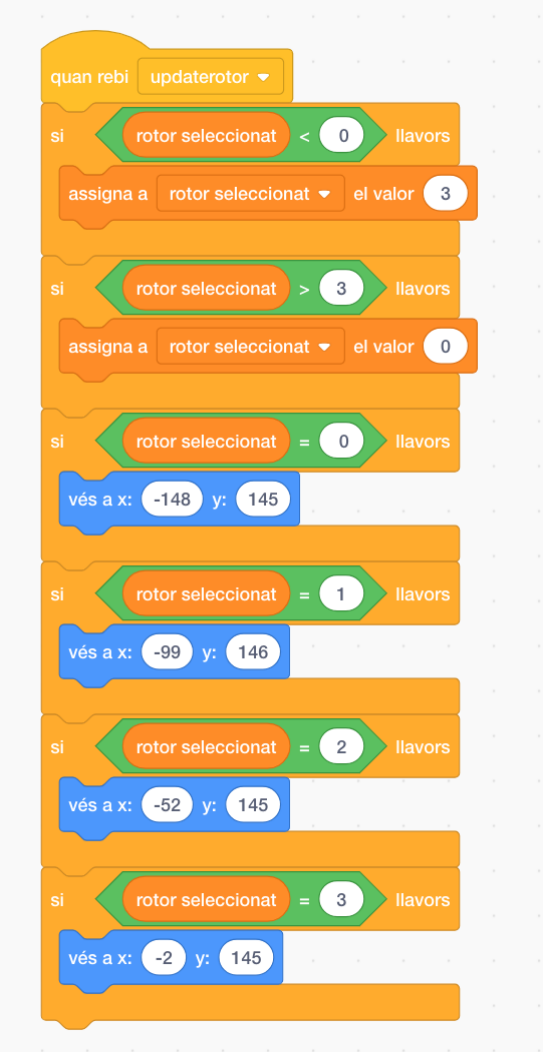



**Imatge 40.** Codi de la llum que il·lumina el text de sortida. *Font pròpia*

Aquests blocs de codi s'inicien quan reben el missatge *lightup*. En el moment en el qual s'enviï aquest missatge, el procés de codificació d'una lletra ja haurà finalitzat, però l'explicaré posteriorment. Quan rebi aquest missatge el personatge, es mostrarà, i s'iniciaran una sèrie d'estructures condicionals. En una d'elles, per exemple, si la variable sortida, de la qual explicaré el càlcul a l'apartat de "codi de l'escenari", és igual a 1, això significarà que la lletra de

sortida serà la A, amb la qual cosa el personatge es mourà cap a les coordenades del teclat on està la A i enviarà el missatge *output1*, que s'utilitzarà per fer aparèixer les lletres de sortida, tal i com s'ha explicat prèviament. Passarà el mateix amb totes les altres estructures condicionals, però, dintre de les accions, canviaran les coordenades i el missatge que s'enviarà.

**Codi de la caixa groga que permetrà canviar la posició dels rotors**




**Imatge 41.** Codi de la caixa groga. *Font pròpia*

Aquest bloc de codi començarà quan es rebi el missatge *updaterotor*, que explicaré després, i començaran a executar-se una sèrie d'ordres que ens permetran canviar la posició del quadre groc per tal que estigui a sobre del rotor que l'usuari esculli. El missatge *updaterotor* s'enviarà quan l'usuari premi la tecla de la fletxa a la dreta o a l'esquerra i es canviarà el valor de la variable *rotor seleccionat*. Per exemple, si el valor de la variable era 0 i l'usuari ha premut un cop la fletxa a la dreta, li haurem de sumar 1 a la variable, però si prem la fletxa a l'esquerra, restarem 1. Abans de tot, cal aclarir que, en informàtica i programació, el primer element sempre serà l'element 0, no l'1. És per això que el rotor I es correspondrà al número 0, el rotor II a l'1, i així successivament.

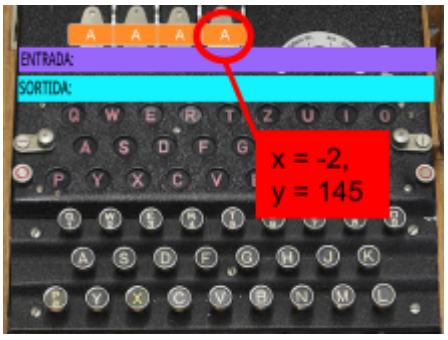
Després de rebre *updaterotor*, hi ha una sèrie d'estructures condicionals. La primera es complirà si el valor del rotor seleccionat és menor que 0. Això passarà si el rotor en el qual estàvem era el número 0 (és a dir, era el rotor I) i premem la fletxa cap a l'esquerra. Haurem de restar 0 menys 1, i dona un valor

negatiu (-1). Per tant, si el valor que tenim és menor que 0, li assignarem a la variable *rotor seleccionat* el valor de 3, cosa que significarà que el rotor en el qual ha d'estar el nostre personatge és el rotor IV.

La segona estructura condicional del nostre codi diu: si la variable *rotor seleccionat* és major que 3, li assignarem el valor de 0. Aquest cas es donarà si estem col·locats al rotor IV (amb la qual cosa, el valor de *rotor seleccionat* és 3), i premem la fletxa a la dreta. Si li sumem 1 al valor actual de la variable (3), obtenim el número 4, però no hi ha un rotor V, amb la qual cosa li donarem el valor de 0 a *rotor seleccionat* per a que es correspongui al rotor I.

A partir d'ara, les quatre estructures condicionals amb les quals ens trobarem descriuran les accions que s'hauran de seguir si escollim el rotor I, II, III o IV (és a dir, si la variable *rotor seleccionat* tindrà el valor 0, 1, 2 o 3, en aquest ordre):

- Si *rotor seleccionat* és igual a 0, és a dir, es correspon al rotor I, haurà d'anar a les coordenades x = -148, y = 145, perquè és en aquest punt en el qual està la variable del rotor I que es mostra a l'escenari.
- Si *rotor seleccionat* és igual a 1, es correspondrà al rotor II i el personatge haurà d'anar a les coordenades x = -99, y = 146.
- Si *rotor seleccionat* és igual a 2, es correspondrà al rotor III i haurà d'anar a les coordenades x = -52, y = 145.
- Si la variable és igual a 3, representarà el rotor IV, amb la qual cosa haurà d'anar a les coordenades x = -2, y = 145.




**Imatge 42.** Codi de la caixa groga. *Font pròpia*

Aquest bloc de codi s'iniciarà quan es premi la bandera verda, és a dir, quan s'iniciï la simulació. Primer de tot, s'enviarà el missatge *uptdaterotor*, que iniciarà els codis explicats prèviament. Després, crearem un bucle repetitiu, i


**Imatge 43.** Codi de la caixa groga. *Font pròpia*

totes les accions que diré a continuació es repetiran per sempre, és a dir, tot el temps que duri la simulació.

Si la fletxa dreta està premuda, llavors s'augmentarà el valor de *rotor seleccionat* en 1, cosa que vol dir que si abans el seu valor era 0 (per tant, estàvem al rotor I) ara el seu valor passarà a ser 1 (amb la qual cosa estarem al rotor II). Després, esperarem, tornarem a enviar el missatge *updaterotor* per iniciar la seqüència explicada a la imatge prèvia i esperarem un temps de 0,2 segons.

Si la fletxa esquerra està premuda, llavors disminuirem el valor de *rotor seleccionat* en 1, envairem *updaterotor* i esperarem 0,2 segons.

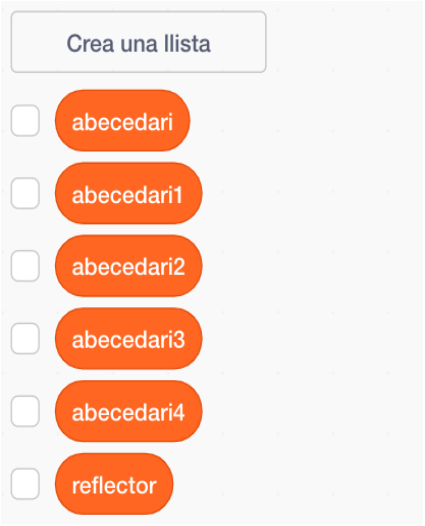
Si la fletxa amunt està premuda, el resultat que voldrem obtenir és que, si al rotor s'indicava la lletra A, ara hi hagi la lletra B. Per fer això, utilitzarem unes altres estructures condicionals. Si la variable *rotor seleccionat* és igual a 0 (amb la qual cosa equivaldrà al rotor I), augmentarem el valor de la variable *rotor1* en 1. Si *rotor seleccionat* és igual a 1, estarem al rotor II, amb la qual cosa haurem d'augmentar el valor de *rotor2* en 1. Si *rotor seleccionat* és igual a 2, augmentarem *rotor3* en 1, i si és igual a 3, augmentarem *rotor4* en 1. Després d'aquestes quatre estructures condicionals, enviarem dos missatges que explicaré després, al codi de l'escenari, que són *checkrotorbounds* i *updaterotordisplay*, i esperarem 0,2 segons.

Si la fletxa avall està premuda, seguirem el mateix procés que abans, però a la inversa, ja que si al rotor hi ha la lletra A i premem la fletxa avall, significa que voldrem obtenir la lletra Z. Si *rotor seleccionat* és igual a 0, disminuirem *rotor1* en 1; si és igual a 1, disminuirem *rotor2* en 1; si és igual a 2, disminuirem *rotor3* en 1; si és igual a 3, disminuirem *rotor4* en 1. Després d'això, enviarem els missatges de *checkrotorbounds* i *updaterotordisplay*, i esperarem 0,2 segons.

**Codi de l'escenari**

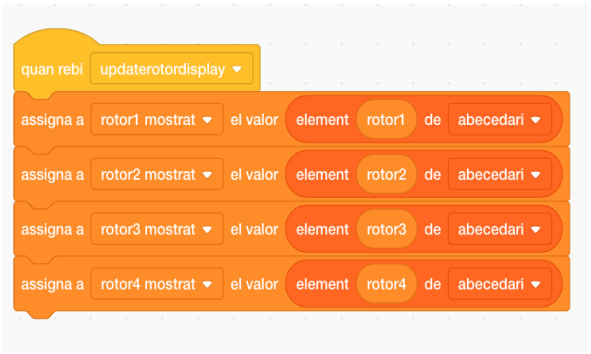

**Imatge 44.** Codi de l'escenari. *Font pròpia*

El codi exposat a continuació s'iniciarà quan es premi la bandera verda. En primer lloc, assignarem a les variables *rotor1*, *rotor2*, *rotor3* i *rotor4*, el valor de 1, perquè, tal i com explicaré després, aquest es correspondrà a la lletra A. Amb això farem que cada vegada que s'iniciï la simulació, sigui el codi que sigui l'establert, tots els rotors es col·loquin amb el codi AAAA. Després, enviarem el missatge *updaterotordisplay*.




**Imatge 45.** Llistes. *Font pròpia*

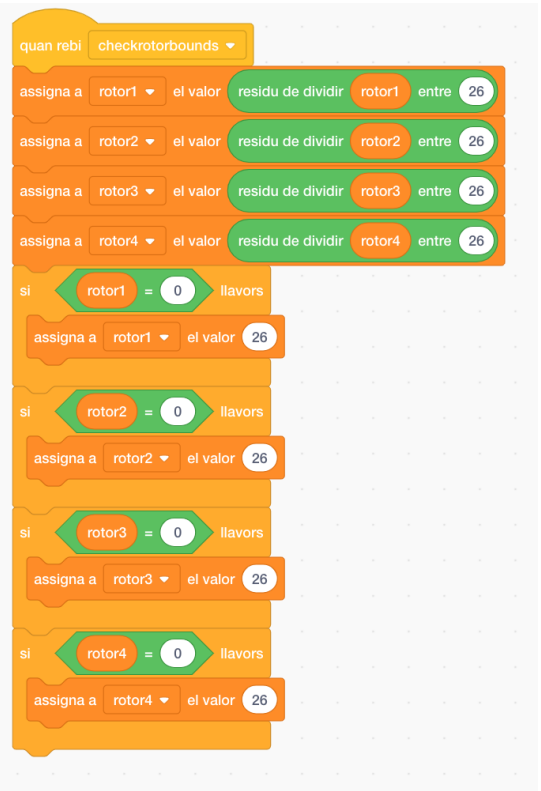
A continuació, fora del codi, haurem de crear diverses llistes. En aquest moment, les variables només estan emmagatzemant nombres, i hem fet diverses operacions amb aquests, però necessitem alguna manera de transformar aquests nombres en lletres. És per això que crearem la nostra pirmera llista: *abecedari*. Els elements d'aquesta llista seran totes les lletres de l'abecedari, i després farem la conversió de nombres a lletres amb la llista. Altres llistes que crearem seran per als rotors i per al reflector.




**Imatge 46.** Codi de l'escenari. *Font pròpia*

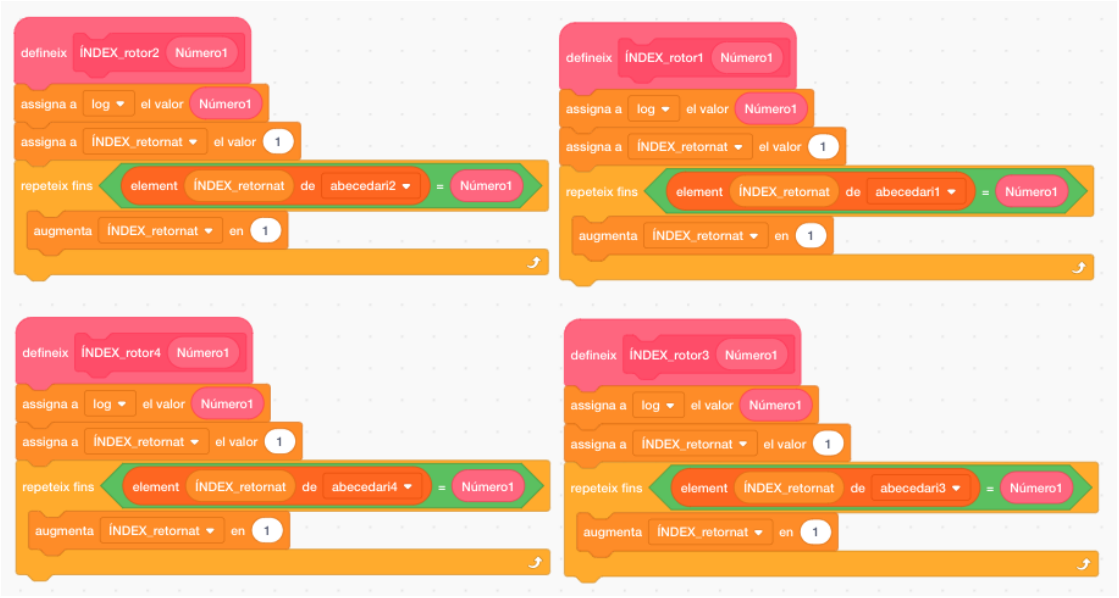
En aquest bloc de codi apareixeran quatre noves variables: *rotor1 mostrat*, *rotor2 mostrat*, *rotor3 mostrat* i *rotor4 mostrat*. A la pantalla d'Scratch, els quatre rotors que apareixeran en realitat seran aquestes variables, però mostrades. Les variables *rotor1*, *rotor2*, *rotor3* i *rotor4* no apareixeran a l'escenari, ja que seran únicament números, però ens serviran per definir aquestes noves variables.

El codi s'iniciarà quan rebi el missatge *updaterotordisplay*, i assignarà als rotors mostrats la lletra corresponent. Per exemple, assignarà a *rotor1 mostrat* el valor de l'element número *rotor1* de la llista abecedari. Si el valor de *rotor1* és 2, l'element número 2 de la llista serà la lletra B, i *rotor1 mostrat* passarà a ser la B. Això es repetirà per als altres rotors.




**Imatge 47.** Codi de l'escenari. *Font pròpia*

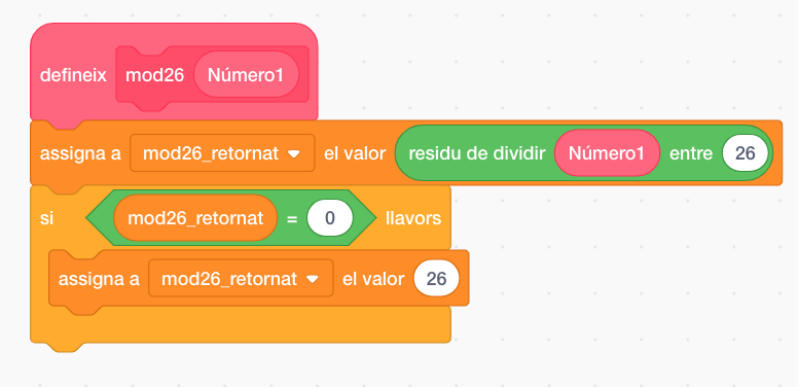
El significat del codi a la imatge superior és el següent: quan es rebi el missatge *checkrotorbounds*, s'iniciarà aquesta seqüència d'instruccions. Li assignarem a *rotor1* el valor del residu de dividir *rotor1* entre 26 (és a dir, *rotor1* mòdul 26), i així successivament amb la resta de rotors. Fem això perquè, si no, amb la fletxa amunt continuaríem augmentant el valor de les variables dels rotors i, per exemple, obtindríem el nombre 27. En fer 27 mòdul 26, obtenim 1, amb la qual cosa estarem escollint la lletra A al rotor. Si el valor de *rotor1* en fer aquesta operació és 0, significa que haurem dividit 26 entre 26, amb la qual cosa li assignarem a *rotor1* el valor 26 (si no, mai no podríem aconseguir la lletra Z, ja que és la número 26 de l'abecedari). Repetirem aquest procés amb els altres rotors.




**Imatge 48.** Codi de l'escenari. *Font pròpia*

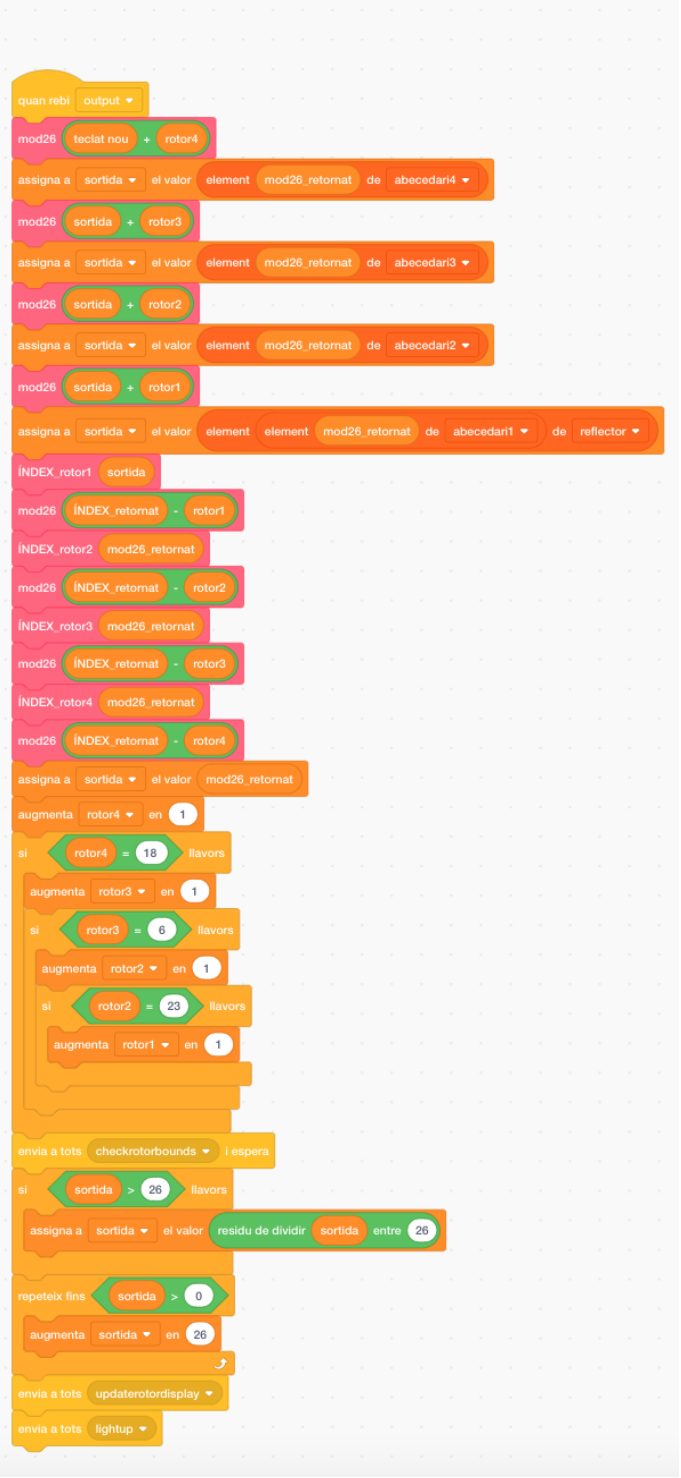
En el següent codi representat, estarem creant els nostres propis blocs. Això serà molt útil per al nostre programa, ja que cada cop que utilitzem un d'aquests blocs, no haurem de tornar a escriure tota la resta codi, cosa que farà que el nostre programa sigui més senzill i comprensible. Per exemple, crearem un bloc que s'anomenarà *ÍNDEX_rotor1*, i establirà una variable, *Número1*. Crearem una nova variable, anomenada *log*, i li assignarem el valor de *Número1*. A continuació, li assignarem a una altra nova variable, *ÍNDEX_retornat*, el valor 1. Després, crearem una altra llista, *abecedari1*, que tindrà els números corresponents a les lletres de l'abecedari, però en un

diferent ordre, i simularà l'abecedari d'un rotor de la màquina Enigma. Finalment, sumarem a *ÍNDEX_retornat* 1 fins que l'element amb el mateix número que el valor de *ÍNDEX_retornat* de la llista *abecedari1* sigui igual a *Número1*. El valor de *Número1* s'explicarà després amb l'últim bloc de codi. Repetirem aquestes instruccions tres vegades més, ja que necessitarem crear en total quatre blocs nous, un per a cada rotor.




**Imatge 49.** Codi de l'escenari. *Font pròpia*

Aquí també crearem un altre bloc, però aquest no serà per als rotors, sinó per a les operacions que necessitarem per transformar una lletra en una altra i codificar els nostres missatges. Definirem un altre bloc, anomenat *mod26* i que tindrà una variable, *Número1*, encara que el valor d'aquesta es determinarà després. També necessitarem crear una nova variable, anomenada *mod26_retornat*. A continuació, assignarem a *mod26_retornat* el valor del resultat de dividir *Número1* entre 26. Després, executarem la següent estructura condicional: si el valor de *mod26_retornat* és igual a 0, li assignarem a la variable el valor de 26 (amb això, estem tornant a fer possible que hi aparegui la lletra Z, ja que és la número 26 a l'abecedari).




**Imatge 50.** Codi de l'escenari. *Font pròpia*

Aquest bloc de codi s'iniciarà quan l'escenari rebi el missatge *output* i, abans de tot, cal esmentar que necessitarem crear una nova variable anomenada *sortida*.

Primer, executarem tots els passos explicats a la creació del bloc *mod26*, i li assignarem a *Número1* el valor de *teclat nou* més *rotor4*. Després, li assignarem a la variable *sortida* el valor de l'element a la llista *abecedari4* que sigui el número que tingui *mod26_retornat* en aquest moment.

Repetirem aquest pas dues vegades més per al rotor III i el rotor II, però en comptes de sumar *sortida* més *rotor4*, utilitzarem el número de rotor adient, i tampoc no utilitzarem la llista *abecedari4*, sinó la que correspongui en aquest cas. Malgrat això, no repetirem el mateix pas per al rotor I. Sí que executarem el bloc *mod26*, i el valor de *Número1* serà *sortida* més *rotor1*, però canviarà el valor que li assignarem a *sortida*. Calcularem quin número és l'element a la llista *abecedari1* que té la posició amb el número de *mod26_retornat*, i agafarem l'element amb aquest número a la llista *reflector*.

A continuació realitzarem una sèrie d'operacions utilitzant els blocs que hem creat prèviament:

- Executarem el bloc *ÍNDEX_rotor1*, i el valor de *Número1* serà *sortida*.
- Realitzarem les operacions del bloc *mod26*, i el valor de *Número1* serà *ÍNDEX_retornat* menys *rotor1*.
- Executarem el bloc ÍNDEX_rotor2, i el valor de *Número1* serà *mod26_retornat*.
- Tornarem a executar el bloc *mod26*, i el valor de *Número1* serà *ÍNDEX_retornat* menys *rotor2*.
- Realitzarem les operacions del bloc ÍNDEX_rotor3, i el valor de *Número1* serà *mod26_retornat*.
- Executarem el bloc *mod26*, i el valor de *Número1* serà *ÍNDEX_retornat* menys *rotor3*.
- Executarem el bloc ÍNDEX_rotor4, i el valor de *Número1* serà *mod26_retornat*.
- Finalment, realitzarem les operacions del bloc *mod26*, i el valor de *Número1* serà *ÍNDEX_retornat* menys *rotor4*.

Després d'aquestes operacions, assingarem a *sortida* el valor de *mod26_retornat*, i augmentarem el valor de *rotor4* en 1. Després de tot aquestes operacions, hem aconseguit codificar una lletra, així que l'últim rotor que té la nostra simulació (és a dir, el rotor IV), haurà de canviar de lletra.

Si *rotor4* és igual a 18, no només voldrem que canviï la lletra del rotor IV, sinó també la del III, així que augmentarem *rotor3* en 1. Si *rotor3* és igual a 6, augmentarem *rotor2* en 1, i si *rotor2* és igual a 23, augmentarem *rotor1* en 1. Després, enviarem el missatge *checkrotorbounds* i esperarem.

Si el valor de la variable *sortida*, que tindrà el valor de la lletra codificada, és major que 26, li assignarem a la variable el valor del residu de dividir *sortida* entre 26. Això vol dir que, si per exemple, el valor de *sortida* és 27, calcularem el residu de dividir 27 entre 26, que serà 1, amb la qual cosa la lletra codificada serà la A. A més, augmentarem *sortida* en 26 unitats si la variable és menor que 0, cosa que ens garantirà que utilitzarem la lletra Z per codificar també.

Finalment, enviarem dos missatges, *updaterotordisplay* i *lightup*, i el programa de la simulació de la màquina Enigma ja haurà acabat.

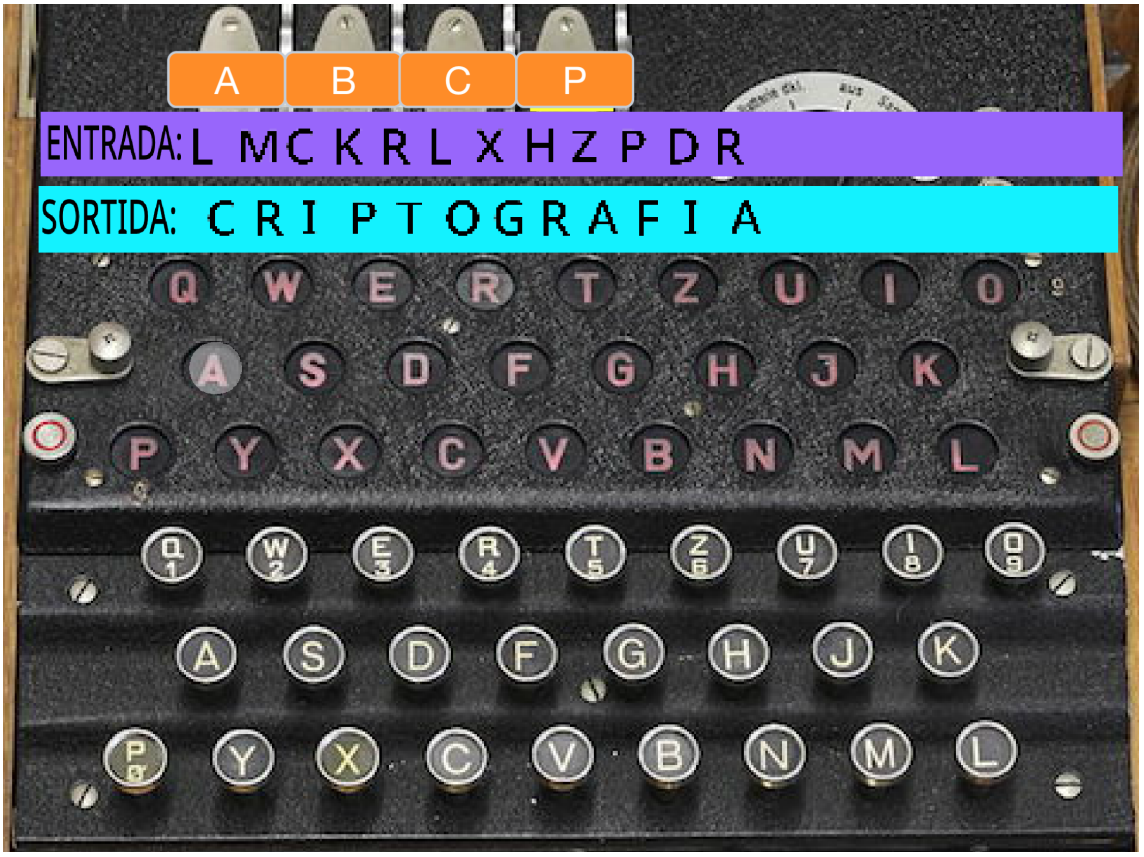
**Funcionament de la simulació de la màquina Enigma amb Scratch**

Per comprovar si el nostre programa funciona, tractarem de codificar i descodificar una paraula. Encriptarem el missatge "criptografia" amb el codi ABCD, per exemple. El resultat que obtenim és el que es mostra a la imatge.


**Imatge 51.** Encriptació de la paraula "criptografia". *Font pròpia*

Primer de tot, he introduït el codi desitjat, i després he escrit la paraula. El resultat ha estat el següent: LMCKRLXHZPDR.

Per desencriptar la paraula, haurem d'establir el mateix codi (ABCD) i escriure el missatge encriptat. Si el programa funciona, obtindrem la paraula "criptografia". El resultat de seguir aquests passos és el que es mostra a la imatge següent.




**Imatge 52.** Desencriptació del missatge obtingut. *Font pròpia*

Podem veure que la desencriptació del missatge ha estat efectiva, ja que hem obtingut la paraula “criptografia” com a resultat. Per tant, el programa és efectiu.

## 6. CONCLUSIONS

Després de realitzar la recerca per dur a terme aquest projecte, he après que la criptografia s'ha utilitzat al llarg de la història per al xifrat de missatges, per tal que només el receptor del missatge que coneguès el codi de xifrat, pogués saber allò que l'emissor volia dir. Però actualment el camp de la criptografia va més enllà del xifrat de missatges, ja que és la base de la ciberseguretat.

Respecte la primera hipòtesi que m'he plantejat en començar el treball puc concloure que sí és possible realitzar un programa d'encriptació i desencriptació amb una aplicació informàtica, l'Scratch en el meu cas, semblant a la màquina Enigma. Aquest programa és capaç d'encriptar un missatge tenint una clau, i el seu funcionament simula de forma informàtica la codificació mecànica de la màquina Enigma.

Respecte la segona hipòtesi que m'he plantejat també puc concloure que l'encriptació de missatges mitjançant aplicacions informàtiques és més efectiva que l'encriptació de missatges manual amb mètodes mecànics. He realitzat una màquina d'encriptació manual, i efectivament el fet d'encriptar i desencriptar missatges és més lent. Tot i així, la màquina Enigma no era del tot manual com la que jo he fet, ja que els rotors giraven automàticament. Però el temps que triguen en girar els rotors sempre és més gran que la velocitat a la que va el programa que he realitzat, ja que aquest, en ser informàtic i no mecànic, dona la informació de forma immediata. Podem pensar que si el missatge és més complex pot arribar a trigar més, però en aquest cas, la màquina mecànica també trigaria més, donat que haurien de girar més cops els rotors.

Finalment, he complert els objectius que em vaig plantejar per tal de validar o refutar les meves hipòtesis, ja que he realitzat una màquina manual de funcionament semblant a la màquina Enigma, un programa informàtic amb l'aplicació Scratch, i he pogut encriptar i desencriptar missatges amb ells, comparant l'eficàcia dels dos. Tot i que puc dir que ha estat difícil programar amb Scratch, donada la complexitat del funcionament de la màquina Enigma

que volia simular, el resultat és molt més ràpid i fàcil d'emprar que la màquina manual que he realitzat.

## 7. AGRAÏMENTS

Agraeixo el suport i ajuda prestada del meu tutor de projecte de recerca, Jaime Morcillo, així com les indicacions que m'ha donat en la realització d'aquest treball.

També agraeixo al programa Estalmat Catalunya i als professors que l'organitzen, per donar l'oportunitat als nois i noies de la meva edat de poder participar-hi. Els hi agraeixo ja que la descoberta del món de la criptografia l'he realitzat a través d'aquest programa.

I finalment agraeixo a la meva germana Laia Marcos i a la meva mare Dori Cañal el suport que d'elles he rebut en tot moment.

## 8. BIBLIOGRAFIA I WEBGRAFIA

Diccionario de la lengua española [en línia]. Madrid: Real academia española, març 2021. [Consultat: 25 de febrer de 2021]. Disponible a <https://dle.rae.es/criptograf%C3%ADa>

Carmona Collado, Luis Miguel. Introducción a la aritmética entera y modular [en línia]. Madrid: Departamento de matemática aplicada de la facultad de informática (U.P.M.), gener 2020. [Consultat: 3 de febrer de 2021]. Disponible a <http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica_modular/criptografia.html>

Gutiérrez, Pedro. Historia de la criptografía [en línia]. Madrid: Genbeta, març 2019. [Consultat: 5 de febrer de 2021]. Disponible a <https://www.genbeta.com/desarrollo/que-es-y-como-surge-la-criptografia-un-repaso-por-su-historia>

Velasco, Juan Jesús. Breve historia de la criptografía [en línia]. Madrid: elDiario.es, 20 de maig del 2014. [Consultat: 5 i 16 de febrer de 2021]. Disponible a <https://www.eldiario.es/turing/criptografia/breve-historia-criptografia_1_4878763.html>

Historia de la criptografía [en línia]. Wikipedia, 15 de febrer de 2021.[Consultat: 5 i 8 de febrer de 2021]. Disponible a < https://es.wikipedia.org/wiki/Historia_de_la_criptograf%C3%ADa>

Historia de la criptografía [en línia]. Barcelona: Binance adademy, 4 d'agost de 2020. [Consultat: 5 de febrer de 2021]. Disponible a <https://academy.binance.com/es/articles/history-of-cryptography>

Venturini, Guillermo. ¿Qué es la criptografía? [en línia]. Tecnología más informática, 2 de octubre de 2020. [Consultat: 5 de febrer de 2021]. Disponible a <https://www.tecnologia-informatica.com/que-es-la-criptografia/>

Código Bacon [en línia]. Wikipedia, 28 de febrer de 2021. [Consultat: 14 de febrer de 2021]. Disponible a < https://es.wikipedia.org/wiki/C%C3%B3digo_Bacon>

El cilindro de Jefferson [en línia]. Granada: Universidad de Granada, gener 2021. [Consultat: 14 de febrer de 2021]. Disponible a <http://www.ugr.es/~anillos/textos/pdf/2012/EXPO-1.Criptografia/02a45.htm>

Cifrado y firmas digitales [en línia]. Granada: Zator Systems, març 2016. [Consultat: 16 de febrer de 2021]. Disponible a <https://www.zator.com/Internet/A6_4.htm>

Aplicaciones de la criptorgrafia [en línia]. València: Universitat de València, gener 2021. [Consultat: 16 de febrer de 2021]. Disponible a <https://www.uv.es/sto/cursos/seguridad.java/html/sjava-10.html>

Velasco, J.J. La máquina Enigma, el sistema de cifrado que puso en jaque a Europa [en línia]. Barcelona: Univesitat Pompeu Fabra, 12 de juliol de 2011. [Consultat: 16 de febrer de 2021]. Disponible a < https://hipertextual.com/2011/07/la-maquina-enigma-el-sistema-de-cifrado-que-puso-en-jaque-a-europa>

En Enigma [en línia]. València: Universitat Politècnica de València, 4 de juliol de 2011. [Consultat: 16 de febrer de 2021]. Disponible a <https://histinf.blogs.upv.es/2011/11/04/2248/>

Enigma, la máquina que cambió el rumbo de la II Guerra Mundial. ABC Cultura [en línia]. Madrid: 3 de juliol de 2020. [Consultat: 16 de febrer de 2021]. Disponible a <https://www.abc.es/cultura/abci-enigma-maquina-cambio-rumbo-guerra-mundial-202006030044_noticia.html?ref=https:%2F%2Fwww.google.com%2F >

Introducció a la criptografia [en línia]. Barcelona: Estalmat Catalunya, setembre 2020 [Consultat: 18 de febrer de 2021]. Disponible a

<https://moodle.feemcat.org/pluginfile.php/2202/mod_resource/content/1/CRIPTOGRAFIA-v1.pdf>

Resumen del análisis de frecuencia [en línia]. Colombia: Crypto badness, 30 de juny de 2009 [Consultat: 19 de febrer de 2021]. Disponible a <http://www.sinfocol.org/2009/06/crypto-badness-100-analisis-de-frecuencias/>

Tumba de Cnumhotep [en línia]. Amigos del antiguo Egipto, 25 de gener de 2021 [Consultat: 27 de febrer de 2021]. Disponible a <http://amigosdelantiguoegipto.com/?page_id=3861>

El cilindro de Jefferson [en línia]. Criptografía, 11 d'abril de 2008 [Consultat: 27 de febrer de 2021]. Disponible a <http://criptografiaurjc.blogspot.com/2008/04/el-cilindro-de-jefferson.html>

¿Cifrado de datos? Imprescindible [en línia]. Bacelona: Universitat La Salle, 12 de febrer de 2015 [Consultat: 22 de març de 2021]. Disponible a <https://blogs.salleurl.edu/es/networking-and-internet-technologies/cifrado-de-datos-imprescindible>

El cuaderno del químico escéptico [en línia]. Blogspot.com, setembre del 2018. [Consultat: 22 i 28 de març de 2021]. Disponible a <http://wolframio1783.blogspot.com/2020/03/una-maquina-enigma-casera.html>

¿Qué es Scratch y para qué sirve? [en línia]. Garaje Imagina, el garaje de la imaginación, agost del 2017 [Consultat: 28 de març de 2021]. Disponible a <https://garajeimagina.com/es/que-es-scratch-y-para-que-sirve/>