

APLICACIÓ DE LA INTEL·LIGÈNCIA ARTIFICIAL EN LA RESOLUCIÓ DEL CUB DE RUBIK

Mar Bosch Castells
Olga Catalán Aragall
Albert Martín García
2n BAT A
Eva Aguiló i Carne Soto

“The good thing about computers is that they do what you tell them to do. The bad news is that they do what you tell them to do.” – Ted Nelson

ÍNDIX

1. Introducció.....	4
2. Intel·ligència Artificial.....	5
2.1. Introducció a la Intel·ligència Artificial.....	5
2.2. Història.....	5
2.3. Característiques d'un agent intel·ligent.....	7
2.4. Categories de la Intel·ligència Artificial.....	7
2.5. Escoles de pensament.....	8
2.6. Aplicacions.....	9
3. El Cub de Rubik.....	10
3.1. Història.....	10
3.2. Mecanisme.....	11
3.3. Relació amb les matemàtiques.....	11
3.4. El número de Déu.....	12
3.5. Notació.....	12
3.6. Mètode de resolució.....	13
3.7. Tipus de cubs.....	13
4. Introducció a la programació.....	14
5. Elaboració del projecte.....	14
5.1. Presentació.....	14
5.2. Elecció del cub.....	15
5.3. Realització d'un programa.....	15
5.3.1. Programa de Windows.....	16
5.3.1.1. Llenguatge usat.....	16
5.3.1.2. Programació.....	16
5.3.2. Aplicació d'Android.....	34
5.3.2.1. Interfície.....	35
5.3.2.2. Funcionament.....	35
6. Perspectives de futur.....	36
6.1. Plantejament del robot.....	36
6.2. Elaboració d'un primer prototip.....	36
7. Conclusions.....	38
8. Agraïments.....	39
9. Fonts d'informació.....	40

1. INTRODUCCIÓ

Vivim en un món on els avenços tecnològics són a l'ordre del dia. En les darreres dècades les màquines han esdevingut eines indispensables en el desenvolupament de la nostra societat. Aquest efecte ha viscut un creixement exponencial donada la progressiva implementació d'autòmats capaços de realitzar tasques sense supervisió activa. Si bé aquests solen estar programats per a executar una activitat concreta i pautada, els avenços en el camp de la Intel·ligència Artificial els han permès adaptar el seu comportament d'acord amb experiències prèvies.

Així doncs, la Intel·ligència Artificial és un concepte molt més proper del que la ciència ficció ens fa creure. Sense anar més lluny, serveis com el del cercador Google[®], Amazon[®] o l'assistent personal Siri han assolit l'hegemonia en el seu sector gràcies a l'ús d'aquesta.

Precisament el fet que afecti de manera inadvertida al nostre dia a dia però alhora sigui tan rellevant en servidors d'ús comú, com els prèviament esmentats, és el que ens va atreure a treballar aquest camp en el nostre Treball de Recerca.

El nostre principal objectiu en l'elaboració d'aquest treball era el d'aproximar-nos al món de la Intel·ligència Artificial mitjançant la creació d'un programa capaç de resoldre un problema lògic popular (el *Cub de Rubik*) i, un cop aquest va estar realitzat, la possible aplicació d'aquest programa.

Per arribar a aquest punt, hem de tenir clars prèviament alguns aspectes com: què és la Intel·ligència Artificial, àrees i formes d'aplicació d'aquesta, què és el *Cub de Rubik*, com podem resoldre'l, funcions i llenguatge usats en programació entre d'altres.

2. INTEL·LIGÈNCIA ARTIFICIAL

2.1. INTRODUCCIÓ A LA INTEL·LIGÈNCIA ARTIFICIAL

La Intel·ligència Artificial o simplement IA és tota demostració feta per una màquina o sistema que posseeix la capacitat d'analitzar una situació i elaborar una resposta envers aquesta.

Aquest camp s'ha desenvolupat amb el desplegament de les tècniques utilitzades en el disseny d'autòmats capaços de realitzar funcions pròximes a les de la intel·ligència humana.

La Intel·ligència Artificial adopta moltes idees, punts de vista i tècniques d'altres disciplines, d'entre les quals hem volgut destacar les Matemàtiques, la Psicologia, la Neurologia i la Computació.

Pel que fa a les Matemàtiques han proporcionat teories de lògica, probabilitat, presa de decisions, i computació. De la mateixa manera, han establert el límit entre el que és decidible i computable, Gödel el 1931 mitjançant el seu Teorema de Turing amb la màquina que, precisament, duu el seu nom. I també la complexitat dels algorismes, mitjançant la Teoria de la NP-completesa realitzada per Steve Cook i Richard Karp als anys 70.

D'altra banda, la Psicologia ha col·laborat amb eines que permeten investigar el funcionament de la ment humana i el desenvolupament de la psicologia cognitiva, la qual actua com un dels fonaments de la Intel·ligència Artificial, ja que defineix teories sobre la conducta i les bases del comportament racional.

I també, la Neurologia que ha contribuït amb l'estudi de les neurones i les connexions que aquestes formen, o la Lingüística que és imprescindible a l'hora de representar i usar el llenguatge natural.

Per acabar, la Computació que ha estat la ciència que ha proporcionat les eines que han permès crear, en un sentit més estricte, la Intel·ligència Artificial.

2.2. HISTÒRIA

Prop del 300 aC, Aristòtil creà el sil·logisme un conjunt d'enunciats que descriuen una part del funcionament de la ment humana i que guarden molta relació amb el pensament lògic, el qual és un pilar de la IA en l'actualitat. Tan sols 50 anys després, Ctesibi d'Alexandria confeccionà la primera màquina autocontrolada: un regulador del

flux d'aigua en funció del temps.

El 1315, Ramon Llull pensà que el raonament podia efectuar-se artificialment pel que creà L'Ars Magna, un estudi basat en la combinatòria que consisteix a proposar un sistema de principis generalíssims, aplicables a totes les ciències, que serveixi d'ajuda per a cercar la veritat i resoldre així els diversos problemes científics.

I no fou fins al 1847, cinc segles després, que es dona un avanç significatiu en la història de la IA quan George Boole establí la lògica proposicional o booleana la qual estengué Gottlob Frege el 1879 creant, així, la Lògica de Primer Ordre.

Als inicis del segle XX, concretament el 1903, Lee De Forest inventà la vàlvula de buit i al 1937 Alan Turing confeccionà la Teoria Computacional, d'altra banda, creà la màquina Turing, una computadora universal capaç, d'entre altres coses, de demostrar que hi ha problemes matemàtics que una màquina no pot resoldre i, tan sols tres anys més tard, Turing i el seu equip construï un dels primers ordinadors electromecànics.

L'any 1943, McCulloch i Pitts presentaren el seu model de neurones artificials, primer estudi que, sense existir el terme d'Intel·ligència Artificial el qual no seria conegut fins al 1956, es reconeix dins del camp. Aquest estudi, mitjançant models computacionals, intentà emular el comportament del cervell humà.

Al llarg dels anys 40 i fins a la primera meitat de la següent dècada, es desenvoluparen una espiral d'aportacions focalitzades en la cibernètica, les ciències de la computació, el raonament formal, etc., d'entre les quals destaca un treball de Turing, *Computing Machinery and Intelligence*, un test capaç de determinar si una màquina era intel·ligent o no que fou durament criticat per no tenir en compte ni la comprensió ni l'intercanvi coherent d'un conjunt de símbols.

En el 1951 s'inventà el transistor, un dels components principals en l'electrònica. És un dispositiu electromagnètic emprat per amplificar i potenciar l'energia de qualsevol objecte elèctric o electrònic. Un dels pares d'aquest invent fou Shockley que, juntament amb Brattain, foren capaços de construir aquest dispositiu.

Tots aquests avenços acabarien ocasionant, en 1956, el naixement oficial de la Intel·ligència Artificial.

Els primers anys de vida de la IA foren molt prometedors, es crearen moltes investigacions amb grans ambicions, fet que atragué molts inversors. Ràpidament arribaren els primers avenços, com ara ELIZA (un robot amb capacitat de parla). Però, tan sols 18 anys després de la seva sortida, el 1974 després que no es complissin les

primeres expectatives i apareguessin els primers problemes greus, tals com ho eren les capacitats limitades dels ordinadors d'aleshores, causaren que la finança d'aquests projectes s'extingís.

Fins que a la dècada dels 80 la Intel·ligència Artificial tornà a posar-se de moda, aquest cop de forma menys precipitada. Aquest nou *boom* desencadenà dos fets importants. El primer, l'aparició dels Sistemes Experts orientats a l'aplicació pràctica dels estudis d'IA, deixant de banda la persecució teòrica de la intel·ligència. I per altra banda, per la creació del projecte *La cinquena generació*, una iniciativa del govern japonès que proposava nous objectius que requerien un canvi del paradigma computacional, que finalment no resultà pròsper.

Fou llavors, el 1987, quan després de la caiguda d'aquest projecte i els problemes que s'experimentaren amb els sistemes experts, es donà una nova crisi de finança d'aquests projectes fins al 1993.

Tres anys més tard, Wos i McCune crearen un programa capaç de resoldre un problema matemàtic que fins llavors cap altre sistema havia pogut resoldre.

2.3. CARACTERÍSTIQUES D'UN AGENT INTEL·LIGENT

En el 1987, Fischles i Firschein descriuen una sèrie de fets propis o atributs d'un agent intel·ligent. Aquests atributs són:

- Té la capacitat d'obtenir coneixement.
- Pot resoldre problemes i, fins i tot, simplificar-ne de difícils.
- És capaç de realitzar operacions complexes.
- Entén, té la capacitat de donar sentit a idees ambigües o contradictòries.
- Planifica, preveu conseqüències i avalua alternatives.
- Coneix els seus propis límits pel que fa a habilitats i coneixement.
- És capaç de distingir tot i les similituds.
- Pot ser original, i fins i tot crear nous conceptes i utilitzar analogies.
- Pot generalitzar.
- Pot percebre i modelar el món exterior.
- Pot entendre i utilitzar el llenguatge i símbols.

2.4. CATEGORIES DE LA INTEL·LIGÈNCIA ARTIFICIAL

- Sistemes que pensen com humans: Simulen xarxes neuronals humanes capaces d'aprendre, prendre decisions i resoldre problemes. Com en seria el

cas de les ciències cognitives.

- Sistemes que actuen com humans: Imiten el comportament humà, són sistemes capaços de realitzar accions i tasques que efectuem nosaltres. Com per exemple, el processament del llenguatge natural, la representació del coneixement o la robòtica.
- Sistemes que pensen racionalment: Imiten o simulen el pensament lògic racional de l'humà, perceben, raonen i actuen. Com la lògica formal, que és la base dels programes intel·ligents o dels sistemes experts (imiten el comportament d'un expert en un camp concret).
- Sistemes que actuen racionalment: Simulen el comportament humà. Com per exemple, una màquina que és capaç d'aprendre arran de les experiències que va tenint, com són els agents experts (entitats capaces de percebre el seu entorn, mitjançant sensors, i actuar havent processat i elaborat una resposta adequada i racional a la situació).

2.5. ESCOLES DE PENSAMENT

- Convencional: També coneguda com a simbòlica-deductiva, es basa en l'anàlisi formal i estadístic del comportament humà enfront diferents situacions.
 - Raonament basat en casos: Ajuden a prendre decisions mentre es resolen problemes.
 - Sistemes experts: Proporcionen una solució a partir del coneixement previ del context i certes regles o relacions.
 - Xarxes bayesianes: Proporcionen solucions a partir d'estadístiques.
 - Intel·ligència Artificial basada en comportaments: Tenen autonomia i poden autoregular-se i controlar-se per millorar.
 - *Smart Process Management*: Proposa una solució a un determinat problema de la mateixa manera que ho faria un especialista en la matèria.
- Computacional: També anomenada *subsimbòlica-inductiva* i implica el desenvolupament i/o aprenentatge interactiu. L'aprenentatge es realitza basant-se en dades empíriques.
 - Màquina de vectors de suport: Sistemes que permeten el reconeixement de patrons genèrics de gran potència.
 - Models ocults de Markov: Aprenentatge basat en la dependència temporal d'esdeveniments probabilístics.
 - Sistemes difusos: Tècniques pel raonament sota incerteses.
 - Computació evolutiva: Aplica conceptes inspirats en la biologia, com la

població, mutació i supervivència del més apte per generar solucions cada cop més millorades, per un problema.

- Xarxes neuronals: Sistemes amb grans capacitats de reconèixer patrons. Les xarxes neuronals creades artificialment són una eina computacional que pretén simular l'estructura i les operacions que realitza el cervell i el sistema nerviós humà en diferents aspectes: la seva capacitat d'aprenentatge, de generalització i de processar ràpidament gran quantitat d'informació en paral·lel. Aquestes xarxes reproduïxen les característiques bàsiques de la sinapsi nerviosa dels éssers vius (vegeu **Analogia entre xarxes neuronals artificials i humanes** a l'Annex).

2.6. APLICACIONS

Al tractar-se d'un producte purament informàtic, la Intel·ligència Artificial s'aplica en multitud d'àmbits com:

- Generació de nombres pseudoaleatoris: Donada la incapacitat de les màquines de generar nombres completament aleatoris, la Intel·ligència Artificial permet al sistema detectar patrons i errors en els algoritmes i trobar una solució.
- Simulació d'entitats intel·ligents: L'aplicació de la Intel·ligència Artificial en la programació de comportaments d'entitats permet dotar-les de realisme i capacitat d'aprenentatge.
- Millora de l'experiència: Es pot fer servir la Intel·ligència Artificial per analitzar les dades introduïdes per l'usuari i processar-les per a millorar l'experiència de l'usuari.
- Reconeixement i traducció de caràcters, textos i parla.
- Reconeixement facial.
- Realitat virtual i processament d'imatge.
- Diagnòstics.
- Recollida de dades.
- Filtració de *spam*.
- Robòtica.

Dins aquests àmbits destaquem els següents usos concrets:

- Creativitat computacional: La creativitat computacional és una especialitat que combina la Intel·ligència Artificial i la psicologia cognitiva. El seu objectiu és crear un sistema capaç de reproduir la creativitat humana i ajudar en l'estudi

d'aquesta. Per a assolir aquest objectiu, s'ha hagut de definir el terme creativitat de manera molt concreta per a poder traduir-lo i programar-lo en una màquina. Els investigadors d'IA Newell, Shaw i Simon, en el seu llibre *The process of creative thinking* van aconseguir fer-ho en 4 criteris:

- La resposta és innovadora i útil.
- La resposta és coherent al problema o repte plantejat.
- La resposta és resultat d'intensa persistència, i no una resposta ràpida que és, generalment, incoherent.
- La resposta clarifica un problema a partir de poca informació.

Arran de crear una màquina capaç d'imitar la creativitat humana i complir els quatre paràmetres anteriors, es va crear la creativitat computacional. Aquesta parteix de la idea que la creativitat no és més que una resposta d'acord amb experiències de situacions prèvies i, per tant, la combinació d'aquestes experiències de manera innovadora es tradueix en respostes coherents i per tant, en aquesta capacitat. Aquest sistema es duu a terme, generalment, a partir d'una simulació del cervell humà anomenada Xarxa Neurològica Artificial.

- Intel·ligència Artificial en jocs: Tota indústria es caracteritza per ser competitiva i és aquesta competitivitat la que porta a les empreses a introduir les últimes tecnologies a les seves cadenes de producció. En el cas de la indústria dels videojocs aquesta, prèviament descrita, es tradueix en l'aplicació de la Intel·ligència Artificial en els seus productes.

3. EL CUB DE RUBIK

3.1. HISTÒRIA

Aquest trencaclosques va ser creat el 1974 pel professor d'arquitectura Ernő Rubik a Budapest, Hongria. Ernő estava obsessionat amb les figures geomètriques i la construcció de les mateixes en 3D intentant que aquestes es poguessin moure internament sense trencar-se. Es va adonar que, si cada un dels blocs girava al voltant d'un centre arrodonit, aquests es podrien moure lliurement, sense trencar-se ni deformar el cub. Així doncs, va posar-se a treballar en el primer prototip, compost per 21 diminuts cubs.

Va intentar-lo vendre a Budapest per, després, fer-lo arribar a la resta del món, però no va tenir gaire èxit. Fins que la companyia *Ideal Toy Corp* els va proposar fabricar un milió d'unitats sota el nom de *Cub Rubik*.

3.2. MECANISME

El cub estàndard 3x3x3 fa 5,7 cm de costat i està format per 26 peces, cada una d'elles connectades entre si a través d'un mecanisme intern. Aquestes peces poden ser o bé els centres (caracteritzats per mantenir una posició estàtica) les arestes o els vèrtexs. Aquest cub està format per sis cares de sis colors diferents (tradicionalment són blanc, groc, blau, vermell, verd i taronja). De manera que els centres tenen un únic color, les arestes en tenen dos i els vèrtexs en tenen tres.

Per una altra banda, cadascun dels sis centres gira en un caragol amb un ressort al cap d'aquests que s'uneixen al centre. Gràcies al caragol, el cub té un grau de tensió, major o menor, autoregurable.

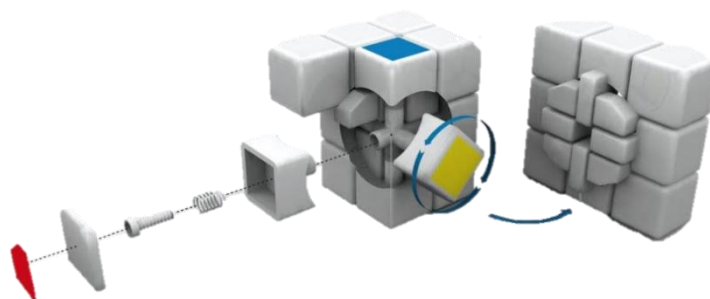


Figura 1. Mecanisme de muntatge d'un cub.

3.3. RELACIÓ AMB LES MATEMÀTIQUES

S'entén com a permutació les diferents posicions en les quals pot trobar-se un cub, és a dir, totes les possibilitats que poden donar-se. Així que (sabent que té vuit vèrtex i dotze arestes) podem calcular el número exacte de configuracions que pot tenir el cub.

Per una banda hi ha vuit maneres de combinar els vèrtexs. Per tant, el nombre de permutacions dels vuit vèrtexs és el factorial de vuit (**8!**) o, el que és el mateix ($8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$), i també hi ha 12 arestes, que com en el cas anterior el nombre de permutacions és el factorial de dotze (**12!** = $12 \times 11 \times 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$). Sempre que les permutacions d'arestes i vèrtexs sigui parells (i com que la meitat de permutacions són parells i les altres imparells) s'ha de dividir entre **2**; per tal d'assegurar-nos que respecti les propietats del cub i no donar lloc a permutacions impossibles.

Alhora, cada vèrtex pot està orientat de tres maneres (**3⁷**). En aquest cas, l'exponent és 7, ja que són 7 els vèrtexs que es poden orientar independentment, el vuitè dependria de l'orientació dels altres.

En el cas de les arestes passa el mateix, cada aresta pot estar col·locada de dues maneres (2^{11}), ja que l'orientació de la dotzena aresta dependrà de les altres onze.

Així que, podem dir que el nombre total de permutacions d'un *Cub de Rubik* es pot expressar de la següent manera:

Permutacions = $\frac{8! \cdot 12! \cdot 3^7 \cdot 2^{11}}{2} = 43\,252\,003\,274\,489\,856\,000$; és a dir, més de 43 trilions de permutacions.

3.4. EL NÚMERO DE DÉU

El número de Déu és el nombre màxim de moviments que es necessiten per resoldre de forma òptima el *Cub de Rubik*. Actualment, aquest número és 20, el que significa que qualsevol cub es pot resoldre amb el número màxim de 20 moviments (o menys) estigui en la posició que estigui.

Per tal de concloure amb aquesta xifra s'han realitzat nombroses investigacions. El primer d'ells fou en el juliol del 1981 per Morwen Thistlethwaite, professor de matemàtiques en la Universitat de Tennessee, que va establir el màxim de moviments en el 52. Thistlethwaite va idear un algoritme capaç de resoldre el cub en un número molt reduït de moviments, però aquest mètode era tan complex que no estava destinat a ser après de memòria o als humans, sinó per la computació i per ser resolt per un ordinador. Les investigacions van anar avançant fins que finalment, el juliol de 2010, Tomas Rokicki, Herbert Kociemba, Morley Davidson i John Dethridge van aconseguir demostrar, treballant juntament amb Google, que va cedir els seus súper ordinadors, que realment el número de Déu és 20.

Per fer-ho, primerament van analitzar les permutacions possibles del *Cub de Rubik* i, al restar totes aquelles configuracions que eren equivalents matemàticament entre elles, es va reduir la xifra de permutacions fins a 55.882.296, que amb l'ajuda dels súper ordinadors es va poder afirmar el número de Déu.

Entre Morwen Thistlethwaite i Tomas Rokicki, Herbert Kociemba, Morley Davidson i John Dethridge n'hi hagué molts d'altres que a poc a poc reduïren el valor d'aquest nombre (vegeu la Figura 54 a l'apartat: **Evolució del número de Déu al llarg dels anys** a l'Annex).

3.5. NOTACIÓ

Per tal d'anomenar les cares i els moviments que es realitzen, utilitzem una notació

estàndard. Aquesta notació està formada per lletres majúscules que són les inicials en anglès del moviment de la cara en qüestió i del símbol prima ['] que ens indica si és sentit antihorari o l'absència d'aquest que indica que és en sentit horari.

- F= Frontal (Front)
- R= Dreta (Right)
- U = Superior (Up)
- B= Posterior (Back)
- D= Inferior (Down)
- L= Esquerra (Left).

Per tant, prenent la cara frontal com a exemple, quan diem F s'ha de fer un quart de volta o un gir 90° en sentit horari de la cara frontal. Si per contra diem F' el gir serà antihorari.

Per acabar, quan posem el número 2 darrere d'una lletra significa que aquell gir és de 180° o de dos quarts de volta.

3.6. MÈTODE DE RESOLUCIÓ

Hi ha diverses maneres i mètodes per resoldre'l; bàsicament consisteixen en un conjunt de moviments que permeten deixar fixes aquelles peces ja col·locades. Com per exemple, un dels mètodes més conegut és el dels principiants, que consta de 7 passos (vegeu les Figures 40-46 a l'apartat: **Mètode de resolució per principiants** a l'Annex).

3.7. TIPUS DE CUBS

Quan es fa referència al *Cub de Rubik*, normalment se'n fa al de 3x3x3, però existeixen moltes variants i modificacions.

Cubs que respecten la seva forma cúbica però el nombre de peces varia. (vegeu Figura 47 a l'apartat: **Tipus de cub** a l'Annex). Existeixen cubs de fins a 11x11x11 o 12x12x12 però aquests no són gaire habituals principalment per la seva grandària, fet que implica que sigui menys pràctic i més laboriós el procés de resolució. De la mateixa manera que el seu preu és major, ja que és un procés més complex i sense tanta demanda.

Per una altra banda, trobem variacions dels cubs anteriors en el nombre de capes (vegeu Figura 48 a l'apartat: **Tipus de cub** a l'Annex). Novament també podem trobar cubs de fins a 3x3x11, tot i que, tal com passava amb els 11x11x11 no són habituals, i

aquestes variacions no només es troben en el cub de 3x3x3 sinó que també les podem trobar en altres cubs com en serien el 4x4x1 o 5x5x6.

També hi ha cubs que no respecten la forma cúbica tradicional, sent piràmides o esferes, d'entre altres, però que tot i això se'ls anomena cubs (vegeu Figura 49 a l'apartat: **Tipus de cub** a l'Annex).

De la mateixa manera que també trobem cubs amb diferents mecanismes, on les seves cares es mouen gràcies a sistemes d'engranatges (vegeu Figura 50 a l'apartat: **Tipus de cub** a l'Annex).

Seguidament, trobem cubs que no es regeixen per colors, sinó per formes, com seria el cas del *Mirror*. Dins d'aquest tipus el més conegut és el del 3x3x3, tot i que també trobem aquestes versions dels altres cubs (vegeu Figura 51 a l'apartat: **Tipus de cub** a l'Annex).

Finalment, trobem altres cubs que són difícils de catalogar o que són variacions peculiars d'algun dels cubs concrets esmentats anteriorment (vegeu Figura 52 a l'apartat: **Tipus de cub** a l'Annex).

4. INTRODUCCIÓ A LA PROGRAMACIÓ

La tasca de programar es basa a donar un conjunt ordenat d'instruccions a un ordinador perquè aquest sigui capaç de resoldre un problema mitjançant un seguit d'operacions. Aquestes instruccions s'escriuen amb un llenguatge propi de la matèria format a partir de símbols (que equivaldrien als components lèxics de la llengua) regits per unes normes precises (que equivaldrien a ser la sintaxi).

Es distingeixen dos tipus de llenguatges de programació: els d'alt nivell i els de baix nivell. El que els divergeix és el fet que els primers no siguin d'ús específic d'una màquina concreta mentre que els segons sí que ho són. Així doncs per tasques que requereixin la creació de programes s'utilitzen els d'alt nivell, tals com els triats en l'elaboració del nostre projecte: el Java i el C++, que posteriorment comentarem.

5. ELABORACIÓ DEL PROJECTE

5.1. PRESENTACIÓ

Com a part pràctica del treball hem desenvolupat un sistema capaç de generar una resposta davant d'un problema complex, tal com seria la resolució d'un *Cub de Rubik*

que, posteriorment ens va fer plantejar la possible aplicació física d'aquesta resolució a través de la robòtica.

Abans d'efectuar qualsevol de les dues pràctiques acabades d'esmentar fou necessari definir certes idees sobre les quals treballar, tals com: el cub que volíem usar, amb quin tipus de suport i llenguatge de programació realitzaríem el projecte o el disseny més eficient de robot, que a continuació presentarem.

5.2. ELECCIÓ DEL CUB

Tal com hem vist, hi ha diversos tipus de cubs amb diferents característiques i complexitats; per tant, hem tingut en compte un seguit de criteris a l'hora de fer la nostra elecció.

En primer lloc, ens va semblar important que el cub en qüestió fos conegut per tothom per tal que es fos conscient del grau de dificultat de resolució i també del temps i esforç que això implica.

Seguidament, era primordial la disponibilitat d'informació pràctica de qualitat sobre aquest, així com el funcionament o els diferents mètodes de resolució, degut a la nostra desconexió d'aquest joc lògic.

Per últim, l'ús d'un cub amb un disseny sense característiques que, tot i ser atractives, dificultessin la seva adaptació a un robot que n'executés la resolució, per exemple: el *Void Cube* és llampanant, el seu disseny sense centre destaca d'entre els altres però els problemes que generaria crear una estructura estable amb un cub amb aquesta peculiaritat no el faria viable per a l'aplicació que descrivim.

Per tant, tenint en compte tots els criteris prèviament esmentats, el millor candidat és el cub 3x3x3, ja que és el que els compleix amb un major grau: és el cub més conegut popularment, el temps aproximat que és tarda en resoldre'l amb el mètode de principiants és al voltant de dos minuts i en tenir els centres facilita la tasca mecànica.

5.3. REALITZACIÓ DEL PROGRAMA

El software encarregat de resoldre el *Cub de Rubik* es divideix en un programa per a Windows que troba els moviments necessaris per a resoldre el cub i una aplicació d'Android que permet copiar un cub físic i enviar-lo al programa principal per a ser solucionat.

Al llarg de l'explicació estan ressaltats amb negreta aquells conceptes clau que cal entendre per l'adequada comprensió de la realització del programa.

5.3.1. PROGRAMA DE WINDOWS

El programa de Windows és l'encarregat de llegir els 9 colors de cadascuna de les 6 cares del cub, recrear un cub virtual a partir d'aquests colors i trobar una seqüència de moviments que permetin retornar aquest cub a l'estat resolt.

5.3.1.1. LENGUATGE USAT

Aquest programa ha estat elaborat amb el llenguatge de programació C++, ja que és un llenguatge de programació de propòsit general molt flexible que permet programar software de tot tipus. Un altre dels motius que ens va portar a emprar aquest llenguatge va ser la seva compatibilitat amb l'experiència prèvia que teníem en programació orientada a objectes. Si bé Java també és un llenguatge molt poderós en aquest sentit, vam decantar-nos pel C++ donada la familiarització que teníem amb l'API de Windows, que permet crear software per a aquest sistema operatiu i explotar-ne totes les seves funcions com la creació i control de finestres clients i servidors per a establir comunicacions a distància com la que permet enviar informació des de l'aplicació d'Android al programa.

5.3.1.2. PROGRAMACIÓ

El programa principal es divideix en la part de comunicació i la part de resolució. La primera s'encarrega del control de la finestra i els seus components i de la comunicació amb el sistema i l'usuari. Mentre que la segona és l'encarregada de tot el relacionat amb la resolució del cub.

5.3.1.2.1. COMUNICACIÓ: API DE WINDOWS

Com tot programa basat en l'**API de Windows** (*Windows application programming interface*), el punt d'entrada és la funció `wWinMain` del fitxer `Main.cpp`. Aquesta funció no conté gaires línies i es limita a inicialitzar el programa, ja que com hem mencionat anteriorment, l'hem estructurat en objectes que s'encarreguen del funcionament de les diverses parts d'aquest.

En aquesta funció inicial es crea una instància de l'**objecte** `MainWnd` que representa la finestra principal i controla el funcionament de tot el programa incloent el **message loop de Windows**. A l'iniciar la finestra s'envia el missatge `WM_CREATE` el qual insta al programa a iniciar la creació de tots els elements de la finestra principal.

5.3.1.2.1.1. INTERFÍCIE

Un cop inicialitzada la pantalla principal, el programa presenta a l'usuari una interfície que permet l'ús del programa.

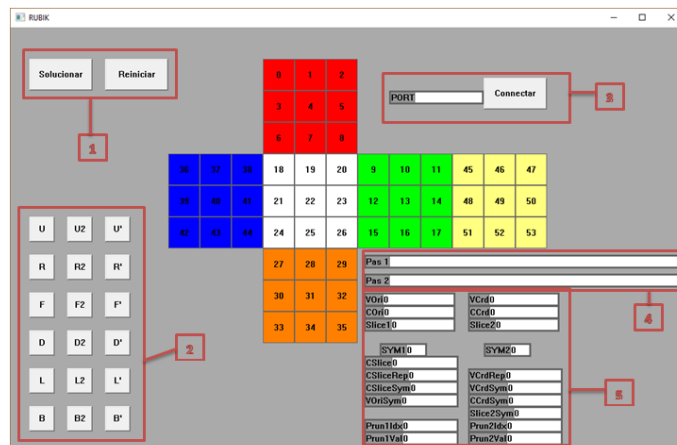


Figura 2. Interfície del programa.

- 1) SOLUCIONAR I REINICIAR: El primer inicia el procés de resolució del cub i el segon permet retornar la simulació del cub a l'estat original. El primer cop que es premi el botó de solucionar trigarà més, ja que s'han de carregar les taules, i encara més si és el primer cop que s'executa el programa, ja que les ha de crear i desar en els arxius *prun1.txt* i *prun2.txt*.
- 2) TAULA DE MOVIMENTS: Permeten executar els 18 moviments possibles.
- 3) CONNEXIÓ: Per a rebre un cub enviat per l'aplicació cal primer inserir un nombre a la caixa de text entre 1000 i 4000 (el mateix que a l'aplicació), prémer el botó Connectar i seguir les instruccions.
- 4) SOLUCIONS: Caixes de text que s'actualitzen al prémer el botó solucionar i mostren les solucions dels dos passos.
- 5) TAULES: Caixes de text que s'actualitzen cada moviment i mostren el valor de les diferents coordenades del cub. Cal haver premut el botó Solucionar al menys un cop per a poder accedir a les taules.
- 6) CUB DE RUBIK: Representació bidimensional del cub.

5.3.1.2.1.2. REPRESENTACIÓ DEL CUB

La representació bidimensional del cub usada no simula un cub tridimensional, sinó que tan sols representa 54 unitats repartides en 6 cares i efectua els moviments movent aquests colors de "quadradet" en "quadradet". Cal

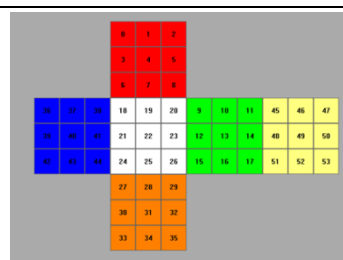


Figura 3. Representació bidimensional del cub.

destacar que els centres no es poden moure, ja que en cas contrari, s'incrementarien els possibles estats del cub i per tant el programa requeriria més temps per a resoldre el cub.

L'ordre de les cares és: vermella, verda, blanca, taronja, blava i groga de manera que se li assigna un valor (c) del 0 al 5 en aquest ordre.

L'ordre dels 9 "quadrats" dins d'una cara va d'esquerra a dreta i de dalt a baix. El nombre de cada "quadrat" (n) es pot obtenir de la manera següent:

$$n = x + 3 * y$$

On x i y són nombres del 0 al 2 que representen la fila i la columna, respectivament, essent $x=0$ i $y=0$ el "quadrat" superior esquerra i $x=2$ i $y=2$ l'inferior dret.

Per tant el nombre de cada "quadrat" (i) en el cub bidimensional es calcula:

$$i = 9 * c + n$$

O, ampliant la fórmula:

$$i = 9 * c + 3 * y + x$$

5.3.1.2.1.3. MOVIMENT DEL CUB

Els 18 moviments que el programa permet fer són els moviments U (Superior), R (Dret), F (Frontal), D (Inferior), L (Esquerra) i B (Darrera). La nomenclatura ve donada per les inicials dels moviments en angles per convenció.

Per a entendre com s'efectuen els moviments cal mirar la cara que es vol girar de manera que el "quadrat" amb $n=0$ es trobi a l'extrem superior esquerra. En aquesta posició es poden fer tres moviments per a la cara que es té al davant: 90° en el sentit de les agulles del rellotge, 180° en qualsevol sentit i 90° en el sentit contrari de les agulles del rellotge. Aquests tres moviments per a cada cara també reben un nombre (s) del 0 al 2, respectivament, de la mateixa manera que també enumerem els moviments amb un nombre (m) del 0 al 17 de la manera següent:

$$m = 3 * c + s$$



Figura 4. Botons corresponents als moviments.

5.3.1.2.1.4. CONNEXIÓ AMB L'APLICACIÓ

Per a rebre un cub de l'aplicació de telèfon cal inserir el mateix nombre a l'espai de "Port" tan al programa com al mòbil. Aquest ha de ser un valor comprés entre 1000 i 4000 ambdós inclosos. Cal també inserir la IPV4 del teu ordinador a l'aplicació. Aquest valor es pot obtenir inserint *ipconfig* al Símbol del sistema.

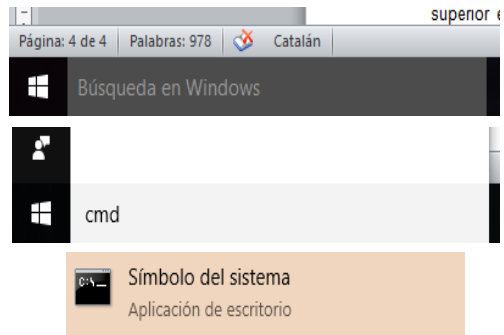


Figura 5. Procediment d'inicialització del Símbol del sistema.

Un cop premut el botó vermell de l'aplicació, s'enviarà el cub del telèfon a l'ordinador via Internet i, si el procés es completa, la representació del cub copiarà els colors de les cares del cub enviat.

5.3.1.2.2. RESOLUCIÓ:ALGORITME DO DOS PASSOS

El problema és el següent: tenim 54 nombres que representen l'estat del cub i volem obtenir una seqüència de moviments que ens permeti resoldre un *Cub de Rubik* en aquest estat.

Aquest és un problema massa complicat per a abordar-lo de cop, per tant apliquem una de les màximes de la programació: "*Divide and conquer*" o "*Divideix i venceràs*". Dividim el nostre problema en una sèrie de reptes menors i els solucionem per separat. Aquesta divisió l'hem efectuat de la manera següent:

- 1) Representar virtualment un cub.
- 2) Transformar els 54 colors en aquesta representació virtual del cub.
- 3) Aplicar moviments a la representació virtual del cub.
- 4) Reduir els possibles estats del cub per a accelerar el procés de resolució.
- 5) Trobar la seqüència de moviments per a resoldre el cub.

5.3.1.2.2.1. REPRESENTACIÓ VIRTUAL DEL CUB

Volem trobar una manera de representar el cub que sigui prou flexible per a poder ser usada en tot el programa per a qualsevol tasca que involucri l'ús de cubs i que alhora no consumeixi gaire memòria i pugui ser modificada eficientment. És per això que fem servir la representació per **coordenades**. Aquest tipus de representació assigna un valor a cada cub depenent de com sigui. En el *Cub de Rubik* tenim 27 cubs dividits en

quatre tipus:

- **Costats:** Aquells que tenen 2 cares i són adjacents a 2 **vèrtexs** i 2 **centres**. En tenim un total de 12 en el cub i els hi assignarem un nombre en base hexadecimal del 0 a B. Cal remarcar que els 4 últims, els costats 8, 9, A, B, pertanyen a una subcategoria de costat anomenada **Slice**.
- **Vèrtexs:** Aquells que tenen 3 cares i són adjacents a 3 **costats**. En tenim un total de 8 al cub i els hi assignarem un nombre del 0 al 7.
- **Centres:** Aquells que tenen 1 cara i són adjacents a 4 **costats** i al **nucli**. En tenim un total de 6 i els hi assignem un valor del 0 al 5 que correspon al valor de la cara mencionat en l'apartat: **5.3.1.2.1.2. Representació del cub.**
- **Nucli:** Centre del *Cub de Rubik* i adjacent als 6 **centres**.

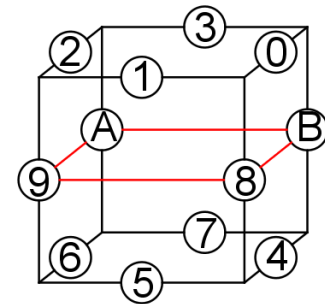


Figura 6. Representació esquemàtica dels costats d'un cub.

L'ordre en què s'han enumerat els diferents cubs dins el seu grup s'anomena **ordre normal**. Aquest és un ordre completament arbitrari, però és el mateix que s'usa en tot el programa. Es recomana fer ús del document *GeoGebra* adjunt (vegeu-lo annexat al CD o Annex virtual, per a la seva visualització es necessita la instal·lació prèvia del programa), ja que permet veure i experimentar amb les diferents posicions que poden prendre els cubs.

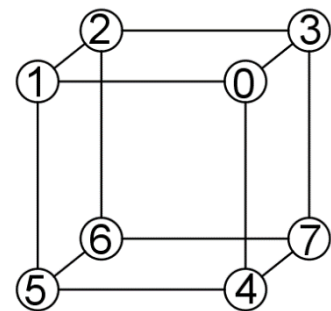


Figura 7. Representació esquemàtica dels vèrtexs d'un cub.

Cal destacar que podem fer aquesta divisió dels cubs en grups, ja que un vèrtex sempre ocuparà una posició de vèrtex i mai d'un costat o un centre, és a dir, un cub d'un cert grup mai no ocuparà una posició d'un cub d'un grup diferent.

Per tant podem fer una representació de les posicions dels cubs d'un grup en forma de taula:

- **Ordre normal dels vèrtexs:**

0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7

Figura 8. Taula d'ordre normal dels vèrtexs.

- **Ordre normal dels costats:**

0	1	2	3	4	5	6	7	8	9	A	B
0	1	2	3	4	5	6	7	8	9	A	B

Figura 9. Taula d'ordre normal dels costats.

La primera fila indica la posició del grup i la segona indica la posició original del cub que ara es troba en aquesta. Per exemple, les dues taules anteriors s'interpreten de la següent manera:

El vèrtex 0 es troba en la posició 0, el vèrtex 1 en la 1, etc. I igual amb els costats.

Però quan el cub es barreja podem trobar taules com aquestes:

- **Taula dels vèrtexs:**

0	1	2	3	4	5	6	7
1	5	2	3	0	4	6	7

Figura 10. Taula dels vèrtexs de F.

Aquesta taula ens diu que a la posició del grup dels **vèrtexs** 0 hi ha el **vèrtex** 1, a la posició 1 el 5, etc.

Particularment aquest és l'ordre en què queden els vèrtexs a l'aplicar el moviment F.

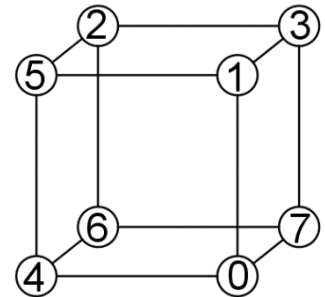


Figura 11. Representació esquemàtica dels vèrtexs de F.

- **Taula dels costats:**

0	1	2	3	4	5	6	7	8	9	A	B
0	9	2	3	4	8	6	7	1	5	A	B

Figura 12. Taula dels costats de F.

Aquesta taula ens diu que a la posició del grup dels **costats** 0 hi ha el **costat** 0, a la posició 1 el 9, etc.

Particularment aquest és l'ordre en què queden els vèrtexs a l'aplicar el moviment F.

Vegem el resultat d'aplicar un moviment F al cub resolt:

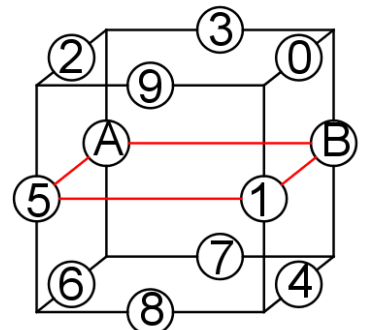


Figura 13. Representació esquemàtica dels costats de F.

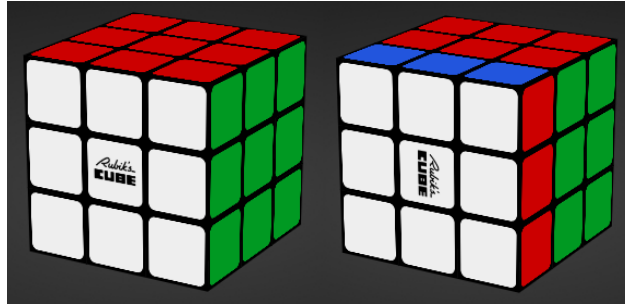


Figura 14. Comparativa entre un cub resolt i F en una representació tridimensional.

Ara tenim una manera de representar estats del cub simple i eficient, però no suficient. Posem per exemple un estat molt curiós del cub anomenat “*superflip*”:

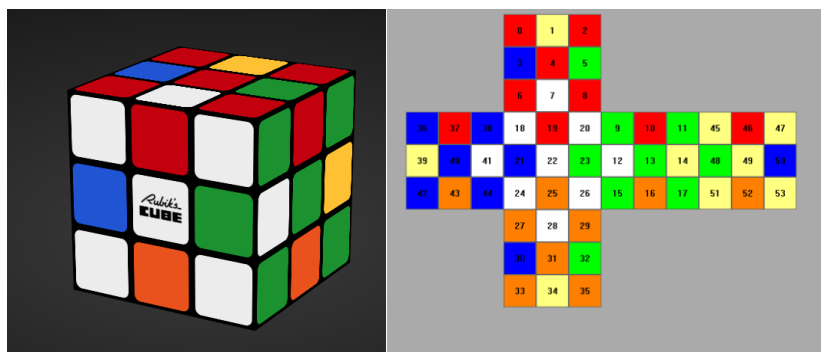


Figura 15. Representació tridimensional i bidimensional del “*superflip*”.

De moment sembla un estat qualsevol del cub, però les taules que hem fet servir per a definir el cub fins ara mostren una peculiaritat:

- **Taula dels vèrtexs:**

0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7

Figura 16. Taula dels vèrtexs del “*superflip*”.

- **Taula dels costats:**

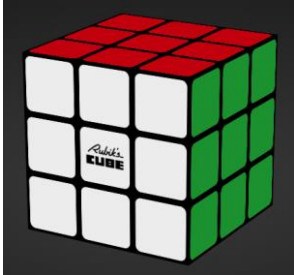
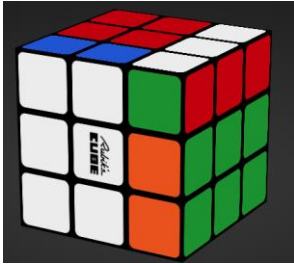

0	1	2	3	4	5	6	7	8	9	A	B
0	1	2	3	4	5	6	7	8	9	A	B

Figura 17. Taula dels costats del “*superflip*”.

Segons les taules, aquest estat del cub és completament igual a l'estat resolt, però òbviament això no és cert. El problema és que hem estat definint l'estat d'un *Cub de Rubik* a partir de les posicions dels cubs en els seus respectius grups sense tenir en compte les **orientacions**. És per això que a partir d'ara definirem l'estat d'un cub en

funció de la **permutació** i l'**orientació**. Amb permutació ens referim al que fins ara anomenàvem estat, és a dir, a la posició que ocupen els cubs en el seu grup. Per tant el que ara anomenàvem **taula** dels **vèrtexs** i **taula** dels **costats** ara ho anomenarem **taula** de la **permutació** dels **vèrtexs** i **taula** de la **permutació** dels **costats**, respectivament. i el nou paràmetre: **orientació**; vindrà representat com a **taula** de l'**orientació** dels **vèrtexs** i **taula** de l'**orientació** dels **costats**, tenint així un total de 4 taules o coordenades que representen el cub.

Definirem l'orientació com un dels possibles estats que pot prendre un cub en una certa posició. Fixem-nos en l'estat del **vèrtex 0** a l'aplicar consecutivament els moviments F i R:

MOVIMENTS FETS	ESTAT DEL VÈRTEX 0
<p>Cap</p> <p>Tenim en cub inicial. El vèrtex 0 es troba en l'orientació correcta, per tant està en orientació 0.</p>	
<p>F i R</p> <p>El vèrtex 0 torna a estar al seu lloc, però on abans hi havia la cara blanca ara hi ha la verda, on hi havia la verda hi ha la vermella i on hi havia la vermella hi ha la blanca. Direm que està en orientació 1.</p>	
<p>F,R,F i R</p> <p>El vèrtex 0 torna a estar al seu lloc un altre cop, però on abans hi havia la cara blanca ara hi ha la verda, on hi havia la verda hi ha la vermella i on hi havia la vermella hi ha la blanca. El procés s'ha repetit però si mires el vèrtex 0 no és igual ni a l'orientació 0 ni a la 1, per tant és orientació 2.</p>	

F,R,F,R,F i R

El **vèrtex 0** és un altre cop al seu lloc, però ara té les cares tal com les tenia a l'inici, per tant el **vèrtex 0** es troba en **orientació 0**.



Figura 18. Taula descriptiva de l'evolució de l'orientació dels vèrtexs.

Per tant, tot vèrtex pot estar en 3 **orientacions** diferents, la 0 és l'estat normal i l'**orientació 1** i 2 corresponen a diferents estats d'aquest en un mateix lloc.

Repetim el procés amb el costat 0 aplicant els moviments R F' i U':

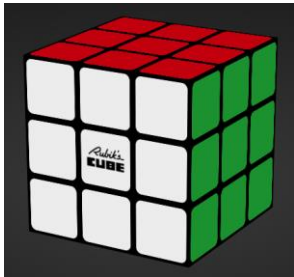
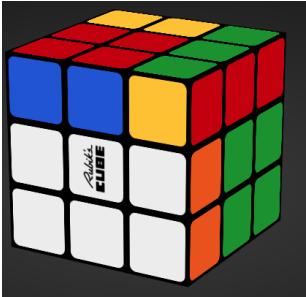

MOVIMENTS FETS	ESTAT DEL COSTAT 0
<p>Cap Tenim el cub inicial, per tant el costat 0 es troba en orientació 0.</p>	
<p>R,F' i U' Després de realitzar els moviments tenim el costat 0 al mateix lloc, però on abans hi havia verd ara hi ha vermell i viceversa. Per tant el costat 0 es troba en orientació 0.</p>	
<p>R,F',U',R,F' i U' Hem tornat a realitzar els moviments i el costat 0 es troba en l'estat inicial. Per tant té orientació 0.</p>	

Figura 19. Taula descriptiva de l'evolució de l'orientació dels costats.

Per tant un costat pot estar en 2 **orientacions** diferents, la 0 és l'estat normal i l'**orientació** 1 correspon a l'únic estat possible per a un **costat** en una certa posició a part de l'**orientació** 0.

Ara podem descriure l'estat d'un cub amb 4 taules o, més concretament, amb 4 **coordenades**:

- **Permutació dels vèrtexs** o **VPerm** o **VCrd**.
- **Permutació dels costats** o **CPerm** o **CCrd**.
- **Orientació dels vèrtexs** o **VOri**.
- **Orientació dels costats** o **COri**.

A partir d'ara farem servir les abreviatures de les coordenades per a referir-nos a aquestes, així que quan parlem de **VPerm** ens referim a la coordenada de la **permutació dels vèrtexs**.

Doncs, apliquem la nova definició d'estat del *Cub de Rubik* als estats resolt i al "**superflip**"

ESTAT RESOLT:

- **Taula de permutacions i orientacions dels vèrtexs.**

POSICIÓ	0	1	2	3	4	5	6	7
PERMUTACIÓ	0	1	2	3	4	5	6	7
ORIENTACIÓ	0	0	0	0	0	0	0	0

Figura 20. Taula de permutacions i orientacions dels vèrtexs de l'estat resolt.

- **Taula de permutacions i orientacions dels costats.**

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
PERMUTACIÓ	0	1	2	3	4	5	6	7	8	9	A	B
ORIENTACIÓ	0	0	0	0	0	0	0	0	0	0	0	0

Figura 21. Taula de permutacions i orientacions dels costats de l'estat resolt.

"**SUPERFLIP**":

- **Taula de permutacions i orientacions dels vèrtexs.**

POSICIÓ	0	1	2	3	4	5	6	7
PERMUTACIÓ	0	1	2	3	4	5	6	7
ORIENTACIÓ	0	0	0	0	0	0	0	0

Figura 22. Taula de permutacions i orientacions dels vèrtexs del "superflip".

- **Taula de permutacions i orientacions dels costats.**

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
PERMUTACIÓ	0	1	2	3	4	5	6	7	8	9	A	B
ORIENTACIÓ	0	1	1	1	1	1	1	1	1	1	1	1

Figura 23. Taula de permutacions i orientacions dels costats del “superflip”.

Ara sí que veiem la diferència entre els dos estats del cub; tots els costats del “*superflip*” tenen orientació 1. Aquest fet es pot apreciar a la foto d’aquest estat si es mira cada **costat** i la posició dels seus colors respecte als **centres** que toquen.

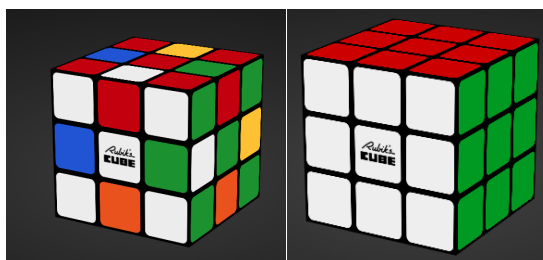


Figura 24. Comparativa entre un cub resolt i el “supeflip” en una representació tridimensional.

Per tant: **Una coordenada representa una part de l'estat del cub independent a les altres coordenades. Les 4 coordenades bàsiques, VOri, COri, VPerm i CPerm defineixen conjuntament un estat de manera única.** Les anomenem coordenades bàsiques, ja que els seus valors s'obtenen a partir de l'estat del cub. Posteriorment definirem un nou tipus de coordenada anomenada **composta** que s'obté a partir de coordenades bàsiques.

Fins ara, en aquest apartat, hem simplificat la definició d'estat del cub:

- La reordenació dels 26 cubs externs al voltant del nucli producte de l'aplicació d'un cert nombre de moviments.

A la definició per coordenades:

- La transposició i reorientació dels cubs entre els espais del seu grup (costat o vèrtex).

Si bé les dues definicions són molt semblants, la seva diferència principal radica en la capacitat de la **definició d'estat del cub per coordenades** de ser **mapejada** a un nombre. **Mapejar**, per definició, vol dir transformar una certa dada en una d'altra de manera reversible i única. Aquest concepte s'aplica al programa per a transformar qualsevol taula que representa una **coordenada** en un nombre que l'identifiqui de

manera que podem transformar la **coordenada** en el nombre i el nombre en la **coordenada** de manera eficient. Aquest procediment l'efectuen les funcions de la classe Mates del programa. Aquest procediment, de forma detallada, es descriu a l'apartat: **Mapejat de coordenades bàsiques** a l'Annex.

Per tant, ara podem definir qualsevol cub amb 4 nombres, que són els valors mapejats de les quatre coordenades bàsiques.

5.3.1.2.2.2. TRADUCTOR D'ESTATS

La transformació de 54 unitats a un estat es realitza de la manera següent:

Primerament emmagatzemem totes les cares que té cada peça, és a dir creem una taula per als costats i vèrtexs com les següents:

Aquestes taules ens indiquen que, per exemple, les 3 cares del **vèrtex 0** són la 8, la 9 i la 20. Aquests 3 nombres són la i , el nombre que identifica cada cara i podem obtenir el seu color si dividim aquest i per 9. Per tant els 3 colors que té el **vèrtex 0** són 0,1 i 2 que corresponen al vermell, verd i blanc respectivament.

VÈRTEX	UNITATS(i)
0	8, 9, 20
1	6, 18, 38
2	0, 36, 47
3	2, 45, 11
4	29, 26, 15
5	27, 44, 24
6	33, 53, 42
7	35, 17, 51

COSTAT	UNITATS(i)
0	5, 10
1	7, 19
2	3, 37
3	1, 46
4	32, 16
5	28, 25
6	30, 43
7	34, 52
8	23, 12
9	21, 41
A	50, 39
B	48, 14

Figura 25. Taules que emmagatzemen les unitats de cada cub.

A partir d'aquestes taules podem trobar quin és el cub que hi ha en una certa posició. Per exemple, si a la posició dels costats 6 tenim els d'un cub amb els colors 50(5) i 39(4) sabem que en aquesta posició hi ha el costat que en un cub resolt ocupa la posició A. Si realitzem aquest mateix procediment amb tots els cubs, podem obtenir

les dues taules de **permutacions**, és a dir, la taula **VPerm** i la taula **CPerm**.

Per a trobar les orientacions cal saber que l'ordre en què els nombres estan ordenats és important. Els colors del vèrtex 0 són 8(0), 9(1) i 20(2) en aquest ordre, ja que quan obtenim els nombres d'un cub en una certa posició, aquests poden estar ordenats de la mateixa manera (indicant, per tant, que es troba en **orientació 0**) o poden estar desplaçats a la dreta 1 o 2 posicions. Per exemple, si hi ha un vèrtex a una certa posició i al trobar els colors d'aquest obtenim 45, 11, 2, veiem que tenim el **vèrtex 3** amb **orientació 2**, ja que els valors estan desplaçats dues posicions a la dreta.

Seguint aquest procediment podem obtenir les dues taules d'**orientacions**, és a dir, la taula **VOri** i la taula **COri**.

5.3.1.2.2.3. APLICACIÓ DE MOVIMENTS

Cal entendre cada estat possible del cub com a resultat d'una successió de moviments. Això és segur perquè si no es pot arribar a un cert estat fent moviments, no hi ha cap altra manera d'arribar-hi. A partir d'aquesta base podem definir qualsevol estat P (ens referim a estats del cub amb lletres majúscules) com a resultat d'una successió dels 18 moviments

PROPIETATS DE LA MULTIPLICACIÓ DE PERMUTACIONS:	
NO COMMUTATIVA:	$F * R \neq R * F$
NO ASSOCIATIVA:	$(F * R) * U \neq F * (R * U)$
IDENTITAT:	$F * I = F$

Figura 26. Taula explicativa de les propietats de la multiplicació de permutacions.

bàsics. Posem per exemple un estat al qual anomenem Q que és el resultat d'aplicar els moviments U, R, F i L sobre un cub resolt al qual ara anomenarem I . Aleshores podem definir Q a partir de la **multiplicació de permutacions**:

$$U * R * F * L = I * U * R * F * L$$

Aquesta fórmula s'interpreta de la manera següent: Comencem amb un cub fet (I) i li apliquem el moviment U , seguit dels moviments R, F i L , en aquest ordre.

Aquesta forma d'aplicar moviments, tot i funcionar bé, no s'ajusta a la nostra definició de **coordenada**, ja que treballem amb coordenades representades per nombres i taules que no admeten moviments i per tant hem d'articular un sistema que ens permeti modificar un cub a partir de les seves **coordenades** sense haver de recrear el

cub per a efectuar el moviment.

Per a fer això cal primer deixar d'entendre els moviments com a tal i veure'ls com a estats resultants de les modificacions més bàsiques que es poden fer a un cub. De manera que ara quan parlem de F o U' no ens referim al moviment Frontal o Superior Invers, sinó a l'estat resultant d'aquest moviment. Per tant els estats F i U' tenen les **coordenades** següents:

- **Taula de permutacions i orientacions dels vèrtexs de F.**

POSICIÓ	0	1	2	3	4	5	6	7
PERMUTACIÓ	1	5	2	3	0	4	6	7
ORIENTACIÓ	1	2	0	0	2	1	0	0

Figura 27. Taules de permutacions i orientacions dels vèrtexs de F.

- **Taula de permutacions i orientacions dels costats de F.**

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
PERMUTACIÓ	0	9	2	3	4	8	6	7	1	5	A	B
ORIENTACIÓ	0	1	0	0	0	1	0	0	1	1	0	0

Figura 28. Taules de permutacions i orientacions dels costats de F.

- **Taula de permutacions i orientacions dels vèrtexs de U'.**

POSICIÓ	0	1	2	3	4	5	6	7
PERMUTACIÓ	3	0	1	2	4	5	6	7
ORIENTACIÓ	0	0	0	0	0	0	0	0

Figura 29. Taules de permutacions i orientacions dels vèrtexs de U'.

- **Taula de permutacions i orientacions dels costats de U'.**

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
PERMUTACIÓ	2	3	0	1	4	5	6	7	8	9	A	B
ORIENTACIÓ	0	0	0	0	0	0	0	0	0	0	0	0

Figura 30. Taules de permutacions i orientacions dels costats de U'.

Per tant, cada moviment **porta una sèrie de cubs des d'una posició fins a una altra**, i com que un estat és una successió de moviments, **un estat és també un moviment**, però més complex. És per això que ara cal veure el que abans enteníem en forma de moviments com a estats resultants d'aquests moviments, tots els estats són successions de moviments, i per tant també es poden definir com a moviments, és a dir, **són translacions de cubs des d'una posició fins a una altra**. Posem per

exemple la **multiplicació de permutacions següent**:

$$F * U'$$

Segons la primera definició que hem donat en aquest apartat, l'equació anterior es pot interpretar com l'aplicació del moviment F sobre I i posteriorment l'aplicació del moviment U sobre l'estat resultant.

Però segons la definició de **multiplicació de permutacions en coordenades**, el que fem és la translació continuada dels cubs. Posem per exemple la **coordenada VPerm**:

POSICIÓ	0	1	2	3	4	5	6	7
I	0	1	2	3	4	5	6	7
I*F	1	5	2	3	0	4	6	7
I*F*U'	3	1	5	2	0	4	6	7

Figura 31. Taules de multiplicacions seqüencials de permutacions en coordenades VPerm.

L'explicació detallada d'aquestes operacions es descriu a l'apartat: **Multiplicació de permutacions en coordenades** a l'Annex.

Un cop trobada la permutació final, es pot trobar l'orientació final sumant tots els canvis d'orientacions que tenen lloc durant les translacions de cubs i trobar el mòdul d'aquesta suma per 3 en el cas dels vèrtexs o 2 en el cas dels costats:

POSICIÓ	0	1	2	3	4	5	6	7
I	0	1	2	3	4	5	6	7
I*F	1	2	0	0	2	1	0	0
I*F*U'	0+1	0+2	0+0	0+0	0+2	0+1	0+0	0+0

Figura 32. Taules de multiplicacions seqüencials de permutacions en coordenades VOri.

Per tant ara distingirem entre dos tipus d'estat: estats bàsics o moviments i estats compostos. Els estats bàsics són els productes directes d'un dels 18 moviments, i els estats compostos són producte de la multiplicació d'estats bàsics. D'aquesta manera tenim dues formes de descriure un estat:

- A partir dels estats bàsics que el componen en forma de multiplicació.
- A partir de quatre les quatre coordenades resultants de la translació de cubs resultant de les translacions parcials dels estats bàsics que componen l'estat compost.

5.3.1.2.2.4. OPTIMITZACIÓ: SIMETRIES

Posem per exemple els estats F i R, segons la definició de **per coordenades** són dos estats completament diferents del cub, però en la realitat podem obtenir F si girem el cub 90° per l'eix vertical, el multipliquem per R o L, i el retornem a la posició inicial. Anomenem a aquest fenomen simetria i ens permet tant reduir el nombre d'estats possibles que caldrà revisar per a resoldre el cub.


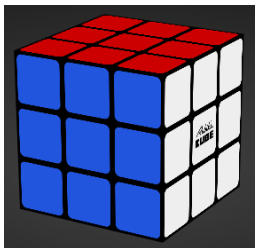
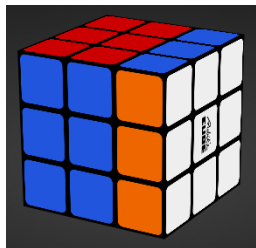

			
F	S⁻¹	S⁻¹*R	S⁻¹*R*S = F

Figura 33. Taula explicativa de les propietats de la multiplicació de permutacions.

Una simetria es pot definir com un moviment de tot el cub, que comporta translacions i reorientacions, pot ser usada en **multiplicacions de permutacions (*)**. Si la simetria usada s'anomena S, l'exemple anterior es pot escriure de la següent manera:

$$F = S^{-1} * R * S = S * L * S^{-1}$$

***Com que les multiplicacions de simetries impliquen un moviment del cub sencer, es compleixen les propietats associativa i commutativa.**

Però no hi ha tan sols una simetria, sinó que n'hi ha tantes com posicions pot adoptar el cub; 6 possibles cares frontals * 4 possibles orientacions que pot prendre cada cara = 48 posicions simètriques diferents. D'aquestes 48, però, només en farem servir les 16 que no alteren l'eix vertical, és a dir que volem que la cara taronja i la vermella sempre siguin o la superior o la inferior.

Totes les simetries són la combinació de les tres simetries bàsiques que poden ser consultades a l'apartat: **Simetries bàsiques** de l'Annex. Per tant tota simetria pot ser representada de la següent manera:

$$S_i = S_{b_0}^{n_0} * S_{b_1}^{n_1} * S_{b_2}^{n_2}$$

On n_0 , n_1 , n_2 són enters entre 0 i 3, 0 i 1 i 0 i 1 respectivament que representen l'estat de cadascuna de les simetries bàsiques S_{B_0} , S_{B_1} i S_{B_2} i i és l'índex de la simetria que s'obté a partir de:

$$i = n_0 + 4 * n_1 + 8 * n_2$$

Una **simetria** inversa és aquella que compleix:

$$S_i * S_i^{-1} = S_0$$

I el seu índex compleix:

$$i = (3 - n_0) + 4 * (1 - n_1) + 8 * (1 - n_2)$$

Ara el cub es pot definir amb 5 valors; els 4 valors obtinguts al **mapejar** les 4 **coordenades** bàsiques i l'índex de la simetria en què es troba el cub.

Un concepte que deriva d'aquest és la **Conjugació de simetries**: Conjuguar una permutació F amb una simetria S_i és aplicar la permutació F sobre la simetria inversa de S_i i tornar a S_0 aplicant sobre la permutació resultant S_i i s'anomena com a F_i .

$$F_i = S_i^{-1} * F * S_i$$

En el programa volem obtenir per a un estat amb els quatre valors mapejats de les coordenades VOri, COri, CPerm i VPerm, uns altres valors VOri' COri' CPerm' i VPerm' que, al conjuguar-se amb la simetria S_i , es transformin en els valors inicials.

En particular volem que els nous valors siguin menors als antics, ja que així reduïm els possibles estats del cub. Per un cert estat pot ser conjugat amb qualsevol de les 16 simetries per a originar 16 estats diferents, però simètrics. Un d'aquests estats es defineix com el **representant** dels altres 15, ja que a la pràctica, els 15 estats són iguals i tan sols requereixen girar el cub per a passar d'un a un altre.

Aquest representant no adopta valors lineals, és a dir, pot ser que un cert nombre pugui ser mapejat a un estat **representant**, però també pot ser que no ho sigui:

ÍNDEX	a	b	c	d	e	F	g	h	i
VALOR	a*16+0	a*16+3	c*16	a*16+14	e*16	c*16+8	a*16+2	e*16+4	c*16+7

Figura 34. Taula comparativa de representants i les simetries amb les que es conjuguen.

Aquesta taula ens diu que una coordenada amb valor mapejat *a* és la representant d'ella mateixa perquè no cal conjuguar-la amb cap simetria. En canvi, si la coordenada té valor *b*, el representant d'aquesta passa a ser *a*, ja que si *a* és conjugada amb la simetria 3, obtenim *b*. En aquesta taula els valors són lletres, ja que no és una taula real. Tan sols serveix per entendre que tota coordenada pot ser expressada mitjançant la conjugació d'una coordenada simètrica a aquesta i una certa simetria. En les caselles de valor apareix l'índex de la coordenada multiplicat per 16, ja que volem emmagatzemar el valor del **representant** i la simetria amb la qual es conjuga en un sol

valor.

Tornant al problema inicial: amb les simetries volíem reduir el nombre d'estats possibles, però de moment no ho hem aconseguit, ja que els valors de les coordenades, tot i ser representants, encara varien entre 0 i el nombre màxim d'aquella coordenada:

REPRESENTANT	0	1	0	0	4	5	1	4	8	0	1	0	5	8	0	16
---------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------

Figura 35. Taula comparativa de representants

Aquest és un exemple que mostra com els valors dels representants no són lineals. Veiem com pels valors 0 i 1, la coordenada es representa a ella mateixa però per al valor 2 (3a casella) torna a ser una conjugació de l'estat 0 i per tant es representa per aquest. Per a solucionar aquest problema assignem un nou valor a cada representant anomenat **representant reduït**:

REPRESENTANT	0	1	0	0	4	5	1	4	8	0	1	0	5	8	0	16
REP. REDUÏT	0	1	0	0	2	3	1	2	4	0	1	0	3	4	0	5

Figura 36. Taula comparativa de representants amb els reduïts.

D'aquesta manera reduïm el màxim d'estats possibles al nombre de coordenades representants.

5.3.1.2.2.5. RESOLUCIÓ: ALGORITME DE DOS PASSOS

Hem basat l'algoritme que resol el cub en les notes creades pel creador de l'Algoritme de Dos Passos, Herbert Kociemba. Aquest algoritme divideix el problema principal de transformar un cub qualsevol en el cub resolt en dos passos. El primer pas té com a objectiu transformar qualsevol estat en un d'altre que compleixi certes característiques:

- Totes les orientacions són 0.
- Tots els costats que pertanyen al grup *Slice* (8, 9, A i B) es troben en una d'aquestes posicions.

Tots els estats que compleixen aquestes condicions s'anomena **estat semiresolt**, i és el segon pas el que s'encarrega de resoldre qualsevol d'aquests estats.

Durant el primer pas tots els 18 moviments són vàlids però durant el segon tan sols es poden realitzar els següents: U,U2,U',R2,F2,D,D2,D', L2 i B2, ja que qualsevol altre altera l'**estat semiresolt**.

El funcionament de l'Algoritme de Dos Passos es basa en què totes les coordenades s'usen tan sols en un pas, i que el cub resolt té totes les coordenades a 0. Partint d'aquest principi podem dir que el primer pas troba la seqüència de moviments per a igualar totes les coordenades associades aquest pas a 0, o el que és el mateix, per a arribar a l'**estat semiresolt**. El pas 2, doncs, troba els moviments necessaris per a igualar la resta de coordenades, aquelles associades al segon pas, a 0. De manera que totes les coordenades coincidiran amb les de l'estat resolt.

Per a realitzar cadascun dels passos es creen prèviament dues taules, una per a cada pas, anomenades taules *prun* (o taules de poda). Aquestes taules tenen tants espais com estats possibles en el seu pas. En cadascun d'aquests espais emmagatzemem el nombre de moviments que calen com a mínim per a arribar l'estat que correspon a aquest espai mòdul 3 (fer un mòdul és trobar el residu de la divisió). És a dir, si calen 10 moviments com a mínim per a arribar a un cert estat, emmagatzemem en la casella corresponent el valor mapejat d'aquest estat 10 mòdul 3 que és igual a 1.

Un cop creades i carregades les taules es procedeix a buscar una solució de la manera següent:

- a) Es mira si l'estat n'és un de resolt, és a dir, es mira si totes les coordenades associades al pas actual és mapejen a 0. Si és així, es procedeix al següent pas (en el cas del pas 1) o es mostra ja la resposta (en el cas del pas 2).
- b) Si no, és un estat resolt se li apliquen tots els moviments possibles en el pas actual i es troben els valors emmagatzemats en les taules *prun* en les caselles corresponents als nous estats i es comparen amb el valor *prun* de l'estat actual. Si el nou és menor vol dir que hem reduït el nombre de moviments que ens calen per a arribar a l'estat resolt, per tant és repeteixen els passos a i b amb aquest nou estat. En cas contrari sabem que no hem reduït el nombre de passos que hem de realitzar, per tant o bé són els mateixos o hem fet un pas que no ens interessa per a resoldre el cub. Per tant, no es procedeix a repetir els passos anteriors.

D'aquesta manera, busca les seqüències de moviments necessàries per a transformar el cub inicial en un **estat semirresolt** i aquest en el cub resolt.

5.3.2. APLICACIÓ D'ANDROID

L'aplicació d'Android permet a l'usuari transferir les cares d'un cub físic al programa de Windows mitjançant Internet.

5.3.2.1. INTERFÍCIE

- 1) TAULA DE CARES: Taula que marca on s'ha de situar cada cara per a la detecció dels colors.
- 2) REPRESENTACIÓ: Quadrat que conté 9 seccions similars a les d'una cara del cub. S'actualitza aproximant els colors als quadrants de la taula de cares a un dels 6 colors del cub. Pot ser bloquejat per a editar manualment els colors fent clic sobre ells.
- 3) IP: caixa de text editable on s'hi ha d'introduir la IPV4 de l'ordinador al qual es volen enviar les dades tal com s'ha explicat a la secció de la interfície del programa de Windows.
- 4) PORT: Caixa de text editable on s'hi ha d'introduir el mateix port que a la del programa de Windows. Aquest pot ser un valor qualsevol entre 1000 i 4000.
- 5) BOTÓ DE CONNEXIÓ: En polsar aquest botó, el telèfon es connecta al programa via internet i envia els colors.

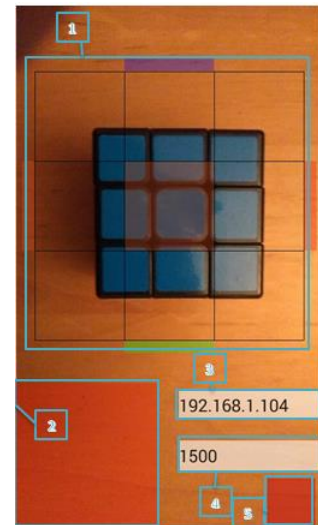


Figura 37. Interfície de l'aplicació.

Tocar qualsevol zona de la taula de cares bloqueja la representació per a modificar-la manualment. Tocar un segon cop la guarda i la prepara per a ser enviada. Un cop realitzar aquest procediment per a cada cara es pot prémer el botó de connexió per a transferir el cub a l'ordinador. A cada costat de la taula de cares s'hi troba el color del costat adjacent per a assegurar un bon angle.

5.3.2.2. FUNCIONAMENT

L'aplicació ha estat programada en Java, ja que és l'únic llenguatge permès per a programar en aquesta plataforma. Android també admet l'addició de parts de codi en C++, però vam decidir no utilitzar aquesta funció per la baixa complexitat de l'aplicació.

L'aplicació manté en tot moment la càmera engegada i analitza de 2 a 10 vegades per segon els colors que coincideixen amb els quadrants de la taula de cares to aproximant-los a un dels colors d'una de les 6 cares.

La representació està desbloquejada a l'inici, i modifica els seus colors en funció dels detectats per la càmera. Al polsar qualsevol zona de la taula la representació es bloqueja i admet la modificació manual. En tornar a polsar la zona de la taula de

costats es desbloqueja i guarda l'estat actual i es prepara per a enviar un cop es polsi el botó de connexió.

6. PERSPECTIVES DE FUTUR

Al acabar l'objectiu principal del nostre treball, la realització d'un programa capaç de resoldre un problema lògic, ens començarem a plantejar possibles línies de treball que seguissin desenvolupant aquest projecte i ens vam centrar en una d'elles: la realització d'un prototip capaç de resoldre mecànicament el cub amb les ordres que rebés del programa que nosaltres mateixos havíem realitzat.

Així doncs, amb l'objectiu de donar-li aquesta altra funcionalitat i extensió a la aplicació inicialment plantejada prosseguirem al plantejament de la construcció d'aquest prototip.

6.1. PLANTEJAMENT DEL ROBOT

En la confecció d'aquest prototip primerament vàrem fer una cerca prèvia de robots ja existents i possibles estructures que ens servissin com a base del nostre robot per tal de després poder extreure les conclusions de quin seria el millor mecanisme en el nostre cas (vegeu **Cerca de robots ja existents** a l'Annex).

Finalment, vam determinar que la millor opció per tal de moure les diferents cares del cub era disposar de sis eixos-motors independents subjectats a la peça central de cada cara del cub, creant així un sistema capaç de moure cada cara sencera del cub i alhora subjectar-lo sense necessitat d'incorporar altres elements externs que podrien comportar encara més dificultats alhora del moviment de les cares.

6.2. ELABORACIÓ D'UN PRIMER PROTOTIP

Un cop havent cercat informació sobre diversos robots ja existents vam prosseguir a elaborar el nostre propi prototip que millor s'adeqüés a la nostra situació.

Aquest, bàsicament, consta de dues parts que poden separar-se en: la superior i la inferior. La superior constituïda per un motor proveït d'un eix per tal de transmetre la rotació d'aquest a la cara superior del cub i la seva estructura que serveix de suport, de la qual parlarem més endavant. I d'altra banda, la part inferior conté cinc motors que controlen les cares: inferior, dreta, esquerra, frontal i posterior (vegeu **Prototip** a l'Annex).

Tal com hem dit, cada cara està unida pel seu centre a un motor mitjançant un eix, aconseguint així una rotació independent de cada cara i per tant un funcionament més simple, ja que les orientacions del cub no varien en cap moment.

Pel que fa a l'estructura, el material triat ha estat principalment la fusta, ja que permet crear una base estable i que s'adapti completament als requisits necessaris.

Respecte al mecanisme per unir el centre de cada cara del cub amb els motors s'ha incorporat al cub una peça a la part central de manera fixa i que encaixa amb l'eix motor. Un sistema que ens permet extreure el cub del mecanisme i després tornar-lo a introduir dins del robot. Aquest últim punt és de vital importància per tal que aquest robot sigui pràctic; si el cub no es pogués extreure perdria fiabilitat, ja que els moviments a l'hora de desfer el cub els faria el mateix robot i no la persona en qüestió.

Per acabar cal dir que el cub que hem trobat més convenient per aquest projecte no és el convencional, el de 57 mm de costat, sinó el que fa 70 mm, ja que ens permetia una mecanització més fàcil i una exigència més baixa en quant a precisió.

7. CONCLUSIONS

La freqüència d'ús de serveis o d'agents proveïts d'Intel·ligència Artificial és més comuna de la qual hom pot arribar a imaginar-se. Aquesta pràctica està molt més estesa del que la majoria de nosaltres coneixem, tot sovint, no som conscients de que utilitzem tecnologies que n'estan dotades.

Malgrat aquest gran abast, les limitacions computacionals dels ordinadors actuals dificulten la creació de servidors intel·ligents més complexos que els actuals, tot i així, aquesta dificultat sembla que podrà ser resolta pels nous ordinadors quàntics que es creu que poden arribar a permetre l'auge d'aquest camp d'estudi.

Centrant-nos en la nostra experiència, hem pogut veure com l'aplicació de la IA en la resolució d'un problema lògic és eficaç, ja que n'accelera el procés. Les capacitats racionals d'aquests dispositius, capaços de fer deduccions complexes per la ment humana amb un temps molt reduït, com en el cas del nostre programa que és capaç de trobar la resolució d'un cub concret en menys de 0,2 segons.

També volem afegir que hi ha infinites possibilitats de crear un programa per assolir un objectiu concret: l'elecció del llenguatge més adient per unes tasques o altres, la creació de les funcions més òptimes, etc. Però, creiem que l'adaptació correcta d'aquest problema a les altes capacitats que ens ofereix la computació moderna és el més important per aconseguir eficiència i funcionalitat en el nostre producte resultant.

Per últim, la IA no és una ciència teòrica sinó que té multiplicitat de funcionalitats pràctiques que més enllà de tasques computacionals poden tractar-se de mecàniques, tal i com hem demostrat amb la realització del nostre prototip.

8. AGRAÏMENTS

Volem agrair especialment l'atenció de Meritxell Gonzalez que, treballant en aquest camp, ens va introduir en el món de la Intel·ligència Artificial.

Volem donar les gràcies a les nostres tutores del Treball de Recerca, l'Eva Aguiló i la Carme Soto, que ens han aconsellat i acompanyat durant tot el procés. També als nostres pares i familiars que no han deixat de recolzar-nos en cap moment i, per últim, a totes les persones que ens han ajudat a realitzar aquest treball.

9. FONTS D'INFORMACIÓ

ABC. (10 d'octubre, 2010). El cubo de Rubik, al desnudo en 20 movimientos. 23/06/2017, de ABC Lloc web: <http://www.abc.es/20100810/tecnologia/20movimientos-cubo-rubik-201008101629.html>

Alvy. (9 d'agost,2010). El número de Dios del Cubo de Rubik es definitivamente 20. 23/06/2017, de Microsiervos Lloc web: <http://www.microsiervos.com/archivo/puzzles-y-rubik/numero-dios-rubik-20.html>

Araceli Sanchis de Miguel, Agapito Ledezma Espino, José A. Iglesias Martínez, Beatriz García Jiménez, Juan Manuel Alonso Weber. Complejidad Computacional. 06/06/2017, de Universidad Carlos III de Madrid Lloc web: <http://ocw.uc3m.es/ingenieria-informatica/teoria-de-automatas-y-lenguajes-formales/material-de-clase-1/tema-8-complejidad-computacional>

BBC. (14 septiembre 2015). Los 10 hitos más importantes en la historia de la inteligencia artificial. 17/05/2017, de BBC Lloc web: http://www.bbc.com/mundo/noticias/2015/09/150914_tecnologia_inteligencia_artificial_hitos_10_turing_asimov_amv

Carlos. (24 de novembre, 2012). No te conformes con el cubo rubik. 15/06/2017, de Zurditorium Lloc web: <http://www.zurditorium.com/no-te-conformes-con-el-cubo-de-rubik>

Carlos Angosto Hernández. Solución sencilla del cubo de Rubik. 23/05/2017, Lloc web: <http://www.rubikaz.com/resolucion.php>

Claudio Gutiérrez. El Teorema de Incompletitud de Gödel. 24/05/2017, de Universidad de Chile Lloc web: <https://users.dcc.uchile.cl/~cguiterr/otros/godel.pdf>

Dr. C. George Boeree. La Neurona. 05/05/2017, de Departamento de Psicología Universidad de Shippensburg Lloc web: <http://webpace.ship.edu/cgboer/genesp/neuronas.html>

Eduardo Sáenz de Cabezón. (19 d'abril, 2017). Matemáticas, el cubo de Rubik y el número de Dios. 20/06/2017, de Derivando Lloc web: <https://www.youtube.com/watch?v=-z4ysmsLI28&vl=es>

El país. (19 de maig, 2014). Cubo de Rubik:¿sabes qué es el número de Dios?. 20/06/2017, de El país Lloc web: https://cincodias.elpais.com/cincodias/2014/05/19/tecnologia/1400490658_663749.htm

!

Enciclopèdia. Ars magna. 06/05/2017, de enciclopèdia.cat Lloc web: <http://www.enciclopedia.cat/EC-GEC-0005416.xml>

Enciclopèdia. Sistema binari. 10/11/2017, de Enciclopèdia.cat Lloc web: <http://www.enciclopedia.cat/EC-GEC-0223065.xml>

Jay Flatland. (11 de gener, 2016). World's Fastest Rubik's Cube Solving Robot. 13/10/2017 Lloc web: <https://www.youtube.com/watch?v=ixTddQQ2Hs4>

Jerish denny. (14 de juliol, 2016). Rubik's Cube Solving Robot. 13/10/2017 Lloc web: <https://www.youtube.com/watch?v=tp4JFz1Pd48>

Jose Varela. (12 agost, 2015). Uno de los padres del transistor; Shockley. 28/04/2017, de A hombros de gigantes. Ciencia y tecnología Lloc web: <https://ahombrosdegigantescienciaytecnologia.wordpress.com/2015/08/12/uno-de-los-padres-del-transistor-shockley/>

Herbert Kociemba. Two-Phase-Algorithm. 13/7/2017, Lloc web: <http://kociemba.org/cube.htm>

Luis Miguel Ardila Rendon. (14 de novembre, 2012). Escuelas del pensamiento (IA). 29/04/2017, Lloc web: http://uniquindioia.blogspot.com.es/2012/11/escuelas-de-pensamiento-ia_14.html

Manuel Alfonseca. La máquina de Turing. 27/04/2017, Lloc web: <http://www.sinewton.org/numeros/numeros/43-44/Articulo33.pdf>

Margaret Rouse. Definition AI (Artificial Intelligence). 06/05/2017, de TechTarget Lloc web: <http://searchcio.techtarget.com/definition/AI>

Matthew Wiebe. (21 d'octubre, 2016). Top 9 ethical issues in artificial intelligence. World Economic Forum, 09/06/2017, Lloc web: <https://www.weforum.org/agenda/2016/10/top-10-ethical-issues-in-artificial-intelligence/>

Miguel Barceló. (2013). En Intel·ligència Ficcio(231). Campus de la UAB, 08193 Bellaterra, Catalunya: Associació Catalana d'Intel·ligència Artificial (ACIA). Lloc web: <https://www.acia.cat/lilibres/barcelo.pdf>

Natalia Maynez. (2017). 10 Tecnologías de Inteligencia Artificial que dominarán el 2017. 06-10-2017, de Adext Lloc web: <https://blog.adext.com/es/tecnologias->

[inteligencia-artificial-2017](#)

Omar Danilo, María del Pilar Rodríguez, Juan David Leyton. (-). Ética e Inteligencia Artificial ¿Necesidad o urgencia?. 10/06/2017, de Universidad Nacional de Colombia Lloc web:

<http://www.iiis.org/CDs2008/CD2008CSC/CISCI2008/PapersPdf/C054TM.pdf>

Pierre Bourdin. (16 de gener, 2017). Robots, intel·ligència artificial i ètica. El Periódico, Opinió. Lloc web: <http://www.elperiodico.cat/ca/noticias/opinio/robots-inteligencia-artificial-etica-articulo-bourdin-5747045>

Professor/a UOC. (24 d'octubre, 2016). Robots, Inteligencia Artificial y Ética. 09/06/2017, de Universitat Oberta de Catalunya Lloc web: <http://informatica.blogs.uoc.edu/2016/10/24/robots-inteligencia-artificial-y-etica/>

Sergio Talens Oliag Curso de programación en C++. 10/11/2017, de UPV Lloc web: <https://www.uv.es/sto/cursos/c++/curso95.pdf>

Tomas Rokicki, Herbert Kociemba, Morley Davidson, John Dethridge. (2010). God's Number is 20. 23/06/2017 Lloc web: <http://www.cube20.org/>

PC. Introducció a les xarxes neuronals. 05/05/2017, de UPC Lloc web: <http://upcommons.upc.edu/bitstream/handle/2099.1/3326/55875-6.pdf?sequence=6>

UPC. Introducción a la inteligencia artificial. 24/04/2017, de UPC Lloc web: <http://www.cs.upc.edu/~bejar/ia/transpas/teoria/1-IA-introduccion.pdf>

Victòria Oliu Subiranas. (27 d'agost, 2010). Comptant en diferents bases. 04/11/2017, de Xtec Lloc web: http://www.xtec.cat/~voliu/sistemes_numeracio/sistema_binari.html

Wilmer y Marlon. (17 d'octubre, 2016). Inteligencia Artificial. 03/05/2017 Lloc web: <http://ipmavinteligenciaartificia.blogspot.com.es/2016/10/en-1987-martin-fischles-y-oscar.html>

Xtec. Binari i decima: conversió. 10/11/2017, de Xtec Lloc web:

<http://blocs.xtec.cat/infotecnorosetaurifiles/2012/09/BINARI-I-DECIMAL.pdf>

ANNEX

ÍNDEX

Annex físic

Analogia entre xarxes neuronals artificials i humanes.....	45
Mètode de resolució per principiants.....	47
Tipus de cubs.....	50
Mapejat de coordenades bàsiques.....	52
Multiplicació de permutacions en coordenades.....	55
Evolució del número de déu al llarg dels anys.....	56
Simetries bàsiques.....	57
Cerca de robots ja existents.....	58
Prototip.....	59

Annex digital

Codi del programa

Codi de l'aplicació

Document *Geogebra*

Aplicació

Programa

ANALOGIA ENTRE XARXES NEURONALS ARTIFICIALS I HUMANES

Les xarxes neuronals humanes són una seqüència de neurones unides, aquestes són les unitats estructurals funcionals fonamentals del sistema nerviós.

Cada una d'aquestes unitats poden dividir-se en:

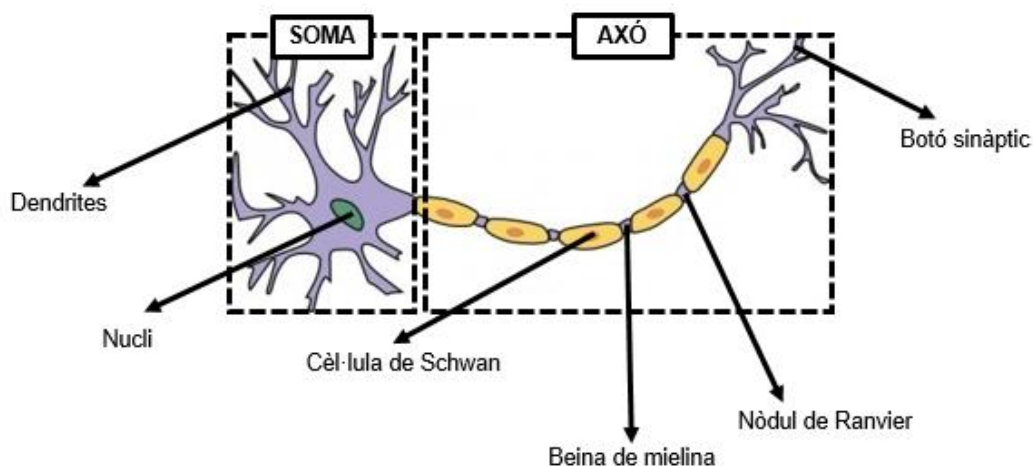


Figura 2. Parts d'una neurona

El soma o cos neuronal és l'espai on es produeixen les reaccions metabòliques corresponents en la sinapsi. En distingim:

- Dendrites: Petites ramificacions que s'estenen de la cèl·lula i que reben l'impuls nerviós procedent d'una altra cèl·lula neuronal (presinàptica).
- Nucli: Orgànu que conté el material genètic en forma de cromosomes.

D'altra banda, hi ha l'axó una extensió per la qual es transmet l'impuls nerviós. En distingim:

- Boto sinàptic: S'uneix a les dendrites d'una altra neurona per a facilitar la sinapsi nerviosa. Conté els neurotransmissors en petites vesícules.
- Beina de mielina: Capes d'un material lipoproteic que envolten l'axó i que permeten la transmissió ràpida i eficient dels impulsos nerviosos.
- Cèl·lula de Schwann: Cèl·lula que, en funció de la grandària del diàmetre de l'axó, s'ocupa de funcions de protecció d'aquesta part de la neurona o bé s'encarrega de formar la beina de mielina.
- Nòduls de Ranvier: Espais o intervals existents entre la beina de mielina que recobreix els axons i que afavoreixen la conducció de l'impuls nerviós.

D'altra banda en funció de la tasca que realitzin en la sinapsi nerviosa o transmissió sinàptica distingim:

- Les neurones sensorials: A diferència de les altres són sensibles a estímuls no neurals, és a dir, són aquelles que capten la informació de l'exterior a través dels diferents sentits.
- Les interneurons: Transmeten l'impuls nerviós entre les neurones sensorials i les motores.
- Les neurones motores: Són capaces d'estimular a les cèl·lules musculars de tot el cos perquè efectuïn una reacció concreta enfront un estímul.

Tot sovint, es troben agrupades en funció de la seva naturalesa, no és habitual trobar-les de forma individual.

El procés de comunicació o transmissió d'informació que es dona entre cèl·lules nervioses, anomenat sinapsi nerviosa, consta de quatre parts:

- Rebre la informació.
- Processar la informació i emetre un impuls nerviós.
- Conduir l'impuls a través de l'axó.
- Comunicar-lo a altres cèl·lules neuronals, tot i que també podrien ser a fibres musculars o a glàndules.

D'altra banda, les xarxes artificials que tracten d'imitar el sistema nerviós humà (com la del model *Parallel Distributed Processing* el qual descriurem a continuació) treballen a partir d'unes unitats de procés, en funció de la seva tasca en el procés en distingirem:

- Les unitats *input*: Caracteritzades per rebre les dades de l'exterior de la xarxa.
- Les unitats ocultes: Reben i emeten els senyals que circulen dins la xarxa.
- Les unitats *output*: Aquelles que envien les dades obtingudes durant el procés a l'exterior.

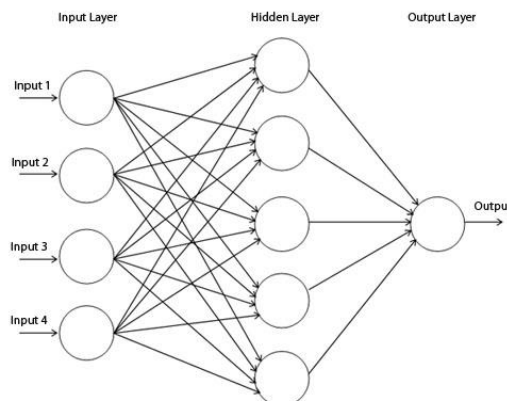


Figura 39. Esquema d'una xarxa neuronal artificial.

La connexió de les unitats de procés o neurones es defineix en funció de la tasca per

la qual s'utilitzi la xarxa. Podria estar connectades a les seves consecutives o bé a les d'una altra capa (entendem com a grup d'unitats que realitzen una funció diferent de la concreta), com sol fer-se a totes les de la xarxa. També existeixen sistemes recurrents, és a dir que transmeten el flux d'informació cap endarrere i que genera, en altres capes, una espècie de memòria que els pot ajudar en l'anàlisi de posteriors estímuls sabent informació d'una anàlisi més avançada.

MÈTODE DE RESOLUCIÓ PER PRINCIPIANTS



1. CREU BLANCA + ARESTES	
<p>Col·locar les arestes blanques amb el seu color corresponent. Localitzar les arestes blanques i portar-les a la cara superior formant una creu blanca i fent que el color continu de les arestes blanques també coincideixi amb el del centre de la cara adjunta.</p>	

Figura 404. Desenvolupament del pas 1.

2. VÈRTEX BLANCS			
<p>Aquest pas pretén acabar la primera cara (la blanca). Per tant, calen col·locar els vèrtex que queden. Arribats a aquest punt ens podem trobar amb quatre situacions diferents, depenent de l'orientació i posició dels vèrtexs.</p>			
<i>Si el vèrtex està a la cara inferior i la part blanca queda a la part dreta de la peça.</i>	<i>Si el vèrtex està a la cara inferior i el blanc queda a la part esquerra.</i>	<i>Si el vèrtex queda a la cara inferior i el blanc queda a la part de sota de la peça.</i>	<i>Si el vèrtex ja està col·locat però està malament orientat</i>
D L D' L'	D' R' D R	R' D2 R D2 F D' F'	ORIENTACIÓ DEL

<p><i>ORIENTACIÓ DEL CUB: la part blanca del vèrtex ha de quedar a la cara frontal.</i></p>	<p><i>ORIENTACIÓ DEL CUB: cal girar la cara inferior de manera que en la columna on hagi d'anar el vèrtex quedi alineat amb la peça malament posicionat, fent que aquesta columna quedi alhora a la dreta de la cara frontal.</i></p>	<p><i>CUB: el vèrtex mal col·locat ha de quedar a la part dreta de la cara i apliquem l'algoritme: R'D'RD, aconseguint així que quedi un dels casos anteriors.</i></p>
---	---	--

Figura 415. Desenvolupament del pas 2.


<h3>3. ARESTES SEGONA CAPA</h3>		
<p>Consisteix en resoldre la segona capa de cub; per fer-ho s'han d'introduir les arestes de la segona capa que, novament les podem trobar en tres situacions diferents.</p> <p><i>ORIENTACIÓ DEL CUB:</i> La aresta mal col·locada i que es vol moure ha de quedar a la cara dreta o, si ja està col·locada, a la columna de la dreta de la cara frontal.</p>		
<p>Està col·locada però malament orientada: R' D R D F D' F' D R' D R D F D' F'</p>		
<p>Per diferenciar els següents dos casos hem de girar la cara inferior fins que el color de l'aresta coincideixi amb el del centre d'una de les cares.</p> <p>Si l'altre color de l'aresta coincideix amb el de la cara de la dreta serà un cas, si ho fa amb el de l'esquerra, en serà un altre.</p>		
<p>Dreta D' R' D R D F D' F'</p>	<p>Esquerra D L D' L' D' F' D F</p>	

Figura 426. Desenvolupament del pas 3.

<h3>4. CREU GROGA</h3>
<p>Fins ara teníem la cara blanca com la cara superior. A partir d'aquest punt, la cara blanca passarà a ser la inferior, és a dir, li donem una volta de 180° al cub i procedirem a fer una</p>

creu groga a la nova cara superior.

Podem trobar-nos amb moltes possibilitats a l'hora d'orientar aquesta cara. A continuació tenim les diferents orientacions que ha d'adoptar el cub en funció del cas que tinguem.

L'objectiu d'aquest pas és arribar a la creu groga, tot aplicant un mateix algoritme de manera repetitiva.





ORIENTACIONS DEL CUB			
Centre groc	"L" invertida a la part superior esquerra	Línia horitzontal	Creu groga
			
$F R U R' U' F'$			

Figura 43. Desenvolupament del pas 4.

5. ARESTES DE LA CREU

ORIENTACIÓ DEL CUB: Cal posicionar-se davant d'una aresta ja col·locada i ben orientada (si no n'hi ha cap s'ha de girar la cara superior fins trobar-nos amb aquesta situació). Un cop localitzada hem de mirar si la cara continua de la dreta també té l'aresta ja col·locada.

Si no és així, s'ha de buscar alguna altra aresta on si es doni aquesta situació. En el cas del que no hi hagi cap s'aplicarà l'algoritme i es tornarà a buscar l'orientació acabada de descriure.



$R'U'R U'R' U^2 R U'$

Figura 447. Desenvolupament del pas 5.

6. VÈRTEXS TERCERA CAPA

ORIENTACIÓ DEL CUB: La cara frontal ha de ser aquella que té un vèrtex ja col·locat (no necessàriament orientat) en l'extrem dret.

Si no n'hi ha cap, aplicar l'algoritme en qualsevol de les cares.

$U R U' L' U R' U' L$



Figura 458. Desenvolupament del pas 6.

7. ORIENTAR ELS VÈRTEXS

ORIENTACIÓ DEL CUB: Cal col·locar com a cara frontal una que tingui el vèrtex superior dret mal orientat. És molt important que, independentment de les repeticions, durant tots els passos la cara frontal sempre sigui la mateixa.

Repetir l'algoritme enllaçant-lo amb U' fins que el cub quedi resolt.

R' D' R D (repetir l'algoritme fins que el vèrtex quedi ven col·locat- normalment són 4 cops-).

En el cas de que hi hagi més d'un vèrtex malament orientat cal enllaçar els algoritmes amb U' (o fins que l'altre vèrtex que està malament orientat es situï al vèrtex dret de la cara frontal) i tornar a aplicar l'algoritme.

Normalment quan s'acaba d'aplicar per últim cop l'algoritme cal fer algun gir a la cara superior perquè es pugui donar per resolt el cub.



Figura 46. Desenvolupament del pas 7.

TIPUS DE CUBS

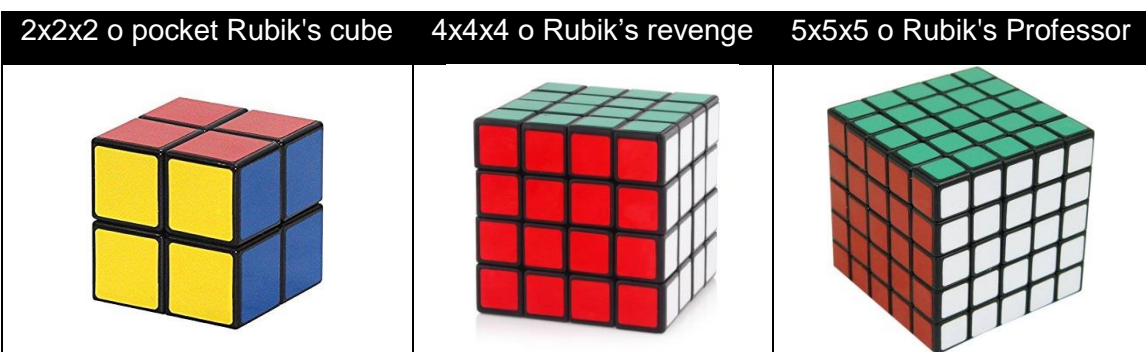


Figura 47. Cubs que respecten la seva forma cúbica.

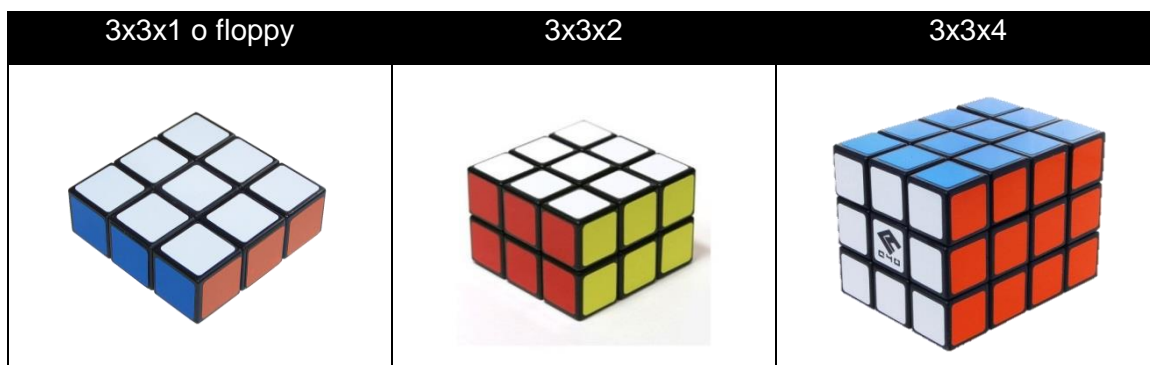


Figura 489. Variacions en el número de capes.

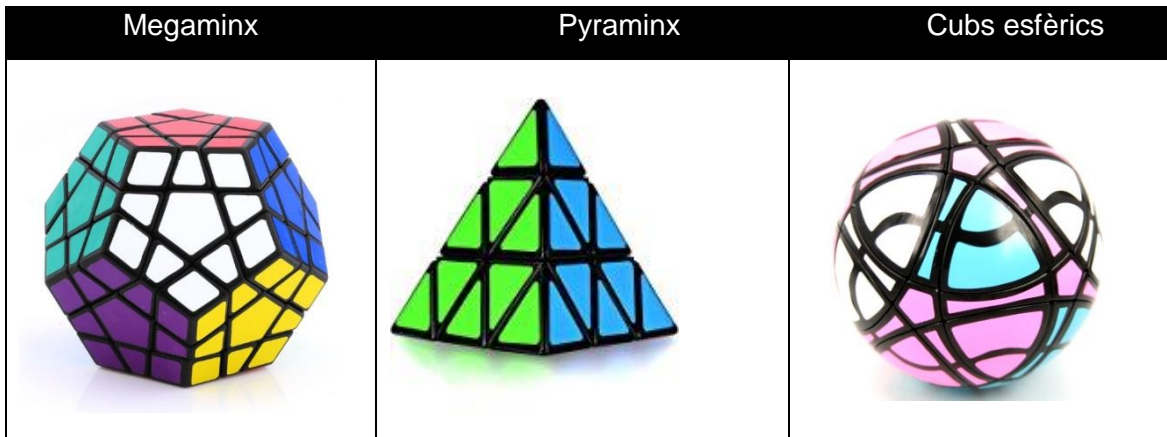


Figura 4910. Cubs que no respecten la forma cúbica.



Figura 5011. Cub amb un mecanisme d'engranatges.

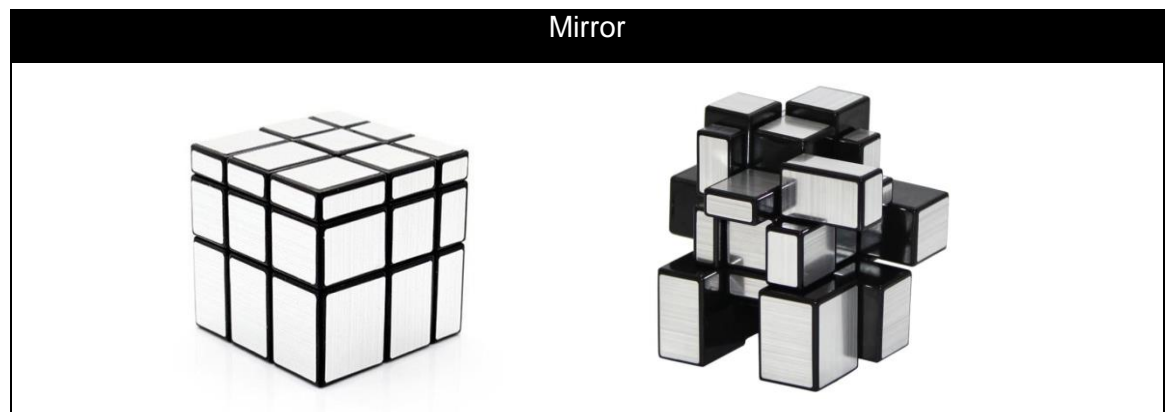
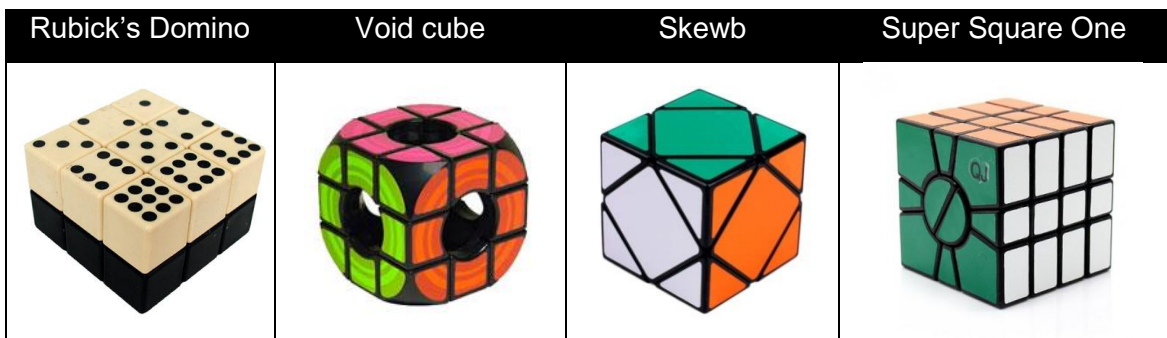


Figura 5112. Cub que no es regeix per colors.



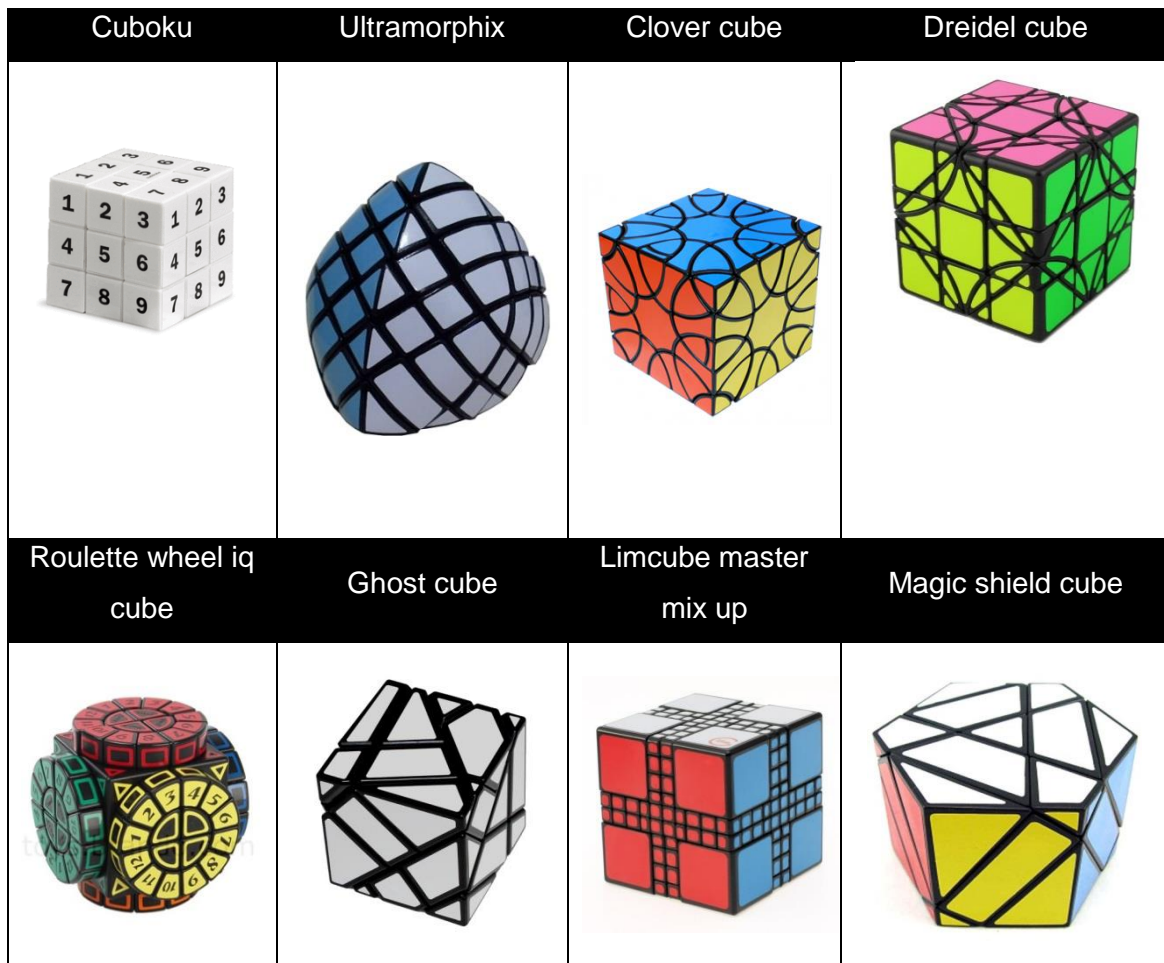


Figura 52. Altres tipus de cubs.

MAPEJAT DE COORDENADES BÀSIQUES

Mapejar VOri a un valor:

Posem per exemple l'estat F. La taula de VOri és la següent:

POSICIÓ	0	1	2	3	4	5	6	7
ORIENTACIÓ	1	2	0	0	2	1	0	0

Per a efectuar la transformació assignem un valor a cada casella i el multipliquem per al valor d'aquesta casella:

POSICIÓ	0	1	2	3	4	5	6	7
ORIENTACIÓ	1	2	0	0	2	1	0	0
CASELLA	729	243	81	27	9	3	1	0
VALOR	729	486	0	0	18	3	0	0

I computem la suma:

$$i = 729 + 481 + 0 + 0 + 18 + 3 + 0 + 0 = 1236$$

Per tant el valor màxim mapejat d'aquesta coordenada és 3^7-1 . Tampoc no es té en compte el valor de l'orientació del vèrtex 7, ja que sabem que la suma de totes les orientacions dels vèrtexs és un múltiple de 3 i per tant aquest últim valor és igual al nombre que caldria sumar al valor mapejat per a ser un múltiple de 3.

Mapejar COri a un valor:

Posem per exemple l'estat F. La taula de COri és la següent:

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
ORIENTACIÓ	0	1	0	0	0	1	0	0	1	1	0	0

Per a efectuar la transformació assignem un valor a cada casella i el multipliquem per al valor d'aquesta casella:

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
ORIENTACIÓ	0	1	0	0	0	1	0	0	1	1	0	0
CASELLA	1024	512	256	128	64	32	16	8	4	2	1	0
VALOR	0	512	0	0	0	32	0	0	4	2	0	0

I computem la suma:

$$i = 0 + 512 + 0 + 0 + 0 + 0 + 32 + 0 + 0 + 4 + 2 + 0 + 0 = 550$$

Per tant el valor màxim mapejat d'aquesta coordenada és $2^{11}-1$. Tampoc no es té en compte el valor de l'orientació del costat B ja, que sabem que la suma de totes les orientacions dels costats és un múltiple de 2 i per tant aquest últim valor és igual al nombre que caldria sumar al valor mapejat per a ser un múltiple de 2.

Mapejar VPerm a un valor:

Posem per exemple l'estat F. La taula de VPerm és la següent:

POSICIÓ	0	1	2	3	4	5	6	7
PERMUTACIÓ	1	5	2	3	0	4	6	7

Per a efectuar la transformació trobem quants vèrtexs amb nombre major al de la casella hi ha en caselles de l'esquerra:

POSICIÓ	0	1	2	3	4	5	6	7
PERMUTACIÓ	1	5	2	3	0	4	6	7
MAJORS	0	0	1	1	4*	1	0	0

*En la posició 4 hi trobem el vèrtex 0. Si comparem el valor en aquesta posició amb els valors de totes les posicions a l'esquerra, és a dir, de les posicions 3, 2, 1, i 0. En aquestes quatre posicions hi trobem quatre valors majors al de la posició 4, que són 3, 2, 5 i 1 en aquest ordre, per tant el valor de majors en a posició 4 és 4.

Li assignem un valor a cada casella i efectuem la multiplicació:

POSICIÓ	0	1	2	3	4	5	6	7
PERMUTACIÓ	1	5	2	3	0	4	6	7
MAJORS	0	0	1	1	4	1	0	0
CASELLA	0	1	2	6	24	120	720	5040
VALOR	0	0	2	6	96	120	0	0

I computem la suma:

$$i = 0 + 0 + 2 + 6 + 96 + 120 = \mathbf{224}$$

Per tant el valor màxim mapejat d'aquesta coordenada és $8! - 1$. Tampoc no es té en compte el valor de l'orientació del vèrtex 0, ja que si sabem les posicions de tots els altres, tan sols ens queda una posició per a l'últim.

Mapejar CPerm a un valor:

Posem per exemple l'estat U. La taula de CPerm és la següent:

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
PERMUTACIÓ	2	3	0	1	4	5	6	7	8	9	A	B

Per a efectuar la transformació trobem quants costats amb nombre major al de la casella hi ha en caselles de l'esquerra:

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
PERMUTACIÓ	3	0	1	2	4	5	6	7	8	9	A	B
MAJORS	0	1	1	1	0	0	0	0	0	0	0	0

Li assignem un valor a cada casella i efectuem la multiplicació:

POSICIÓ	0	1	2	3	4	5	6	7	8	9	A	B
PERMUTACIÓ	3	0	1	2	4	5	6	7	8	9	A	B
MAJORS	0	1	1	1	0	0	0	0	0	0	0	0
CASELLA	0	1	2	6	24	120	720	5040	0	0	0	0

VALOR	0	1	2	6	0	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---	---	---	---	---

I computem la suma:

$$i = 0 + 1 + 2 + 6 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 9$$

Per tant el valor màxim mapejat d'aquesta coordenada és $8! - 1$. Tampoc no es té en compte el valor de l'orientació del costat 0, ja que si sabem les posicions de tots els altres, tan sols ens queda una posició per a l'últim. Tampoc es compten els últims 4 costats, ja que la seva posició és irrellevant en el moment en què aquesta coordenada es fa servir.

Figura 53. Taula de mapejat de coordenades bàsiques.

MULTIPLICACIÓ DE PERMUTACIONS EN COORDENADES

- En multiplicar $I * F$, F porta el cub en la posició 0 a la posició 4, i en multiplicar aquest estat resultant per U' , U' porta el cub en la posició 4 a la posició 4.
- En multiplicar $I * F$, F porta el cub en la posició 1 a la posició 0, i en multiplicar aquest estat resultant per U' , U' porta el cub en la posició 0 a la posició 1.
- En multiplicar $I * F$, F porta el cub en la posició 2 a la posició 2, i en multiplicar aquest estat resultant per U' , U' porta el cub en la posició 2 a la posició 3.
- En multiplicar $I * F$, F porta el cub en la posició 3 a la posició 3, i en multiplicar aquest estat resultant per U' , U' porta el cub en la posició 3 a la posició 1.
- En multiplicar $I * F$, F porta el cub en la posició 4 a la posició 5, i en multiplicar aquest estat resultant per U' , U' porta el cub en la posició 5 a la posició 5.
- En multiplicar $I * F$, F porta el cub en la posició 5 a la posició 1, i en multiplicar aquest estat resultant per U' , U' porta el cub en la posició 1 a la posició 2.
- En multiplicar $I * F$, F porta el cub en la posició 6 a la posició 6, i en multiplicar aquest estat resultant per U' , U' porta el cub en la posició 6 a la posició 6.
- En multiplicar $I * F$, F porta el cub en la posició 7 a la posició 7, i en multiplicar aquest estat resultant per U' , U' porta el cub en la posició 7 a la posició 7.

EVOLUCIÓ DEL NÚMERO DE DÉU AL LLARG DELS ANYS

Juliol del 1981	Morwen Thistlethwaite	53 moviments
Desembre de 1990	Hans Kloosterman	42 moviments
Maig de 1992	Michael Reid	39 moviments
Maig de 1992	Dik Winter	37 moviments
Gener de 1995	Michael Reid	29 moviments
Desembre de 2005	Silviu Radu	28 moviments
Abril de 2006	Silviu Radu	27 moviments
Maig de 2007	Dan Kunkle	26 moviments
Març de 2008	Tomas Rokicki	25 moviments
Abril de 2008	Tomas Rokicki i John Welborn	23 moviments
Agost de 2008	Tomas Rokicki i John Wilborn	22 moviments
Juliol de 2010	Tomas Rokicki, Herbert Kociemba, Morley Davidson i John Dethridge	20 moviments

Figura 54. Taula on es recullen les dates, investigadors i nombre de moviments del número de Déu.

SIMETRIES BÀSIQUES

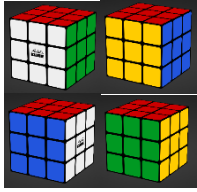
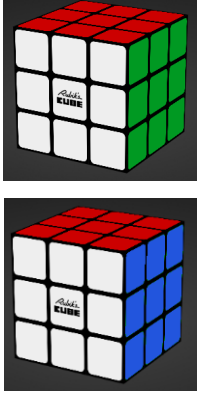
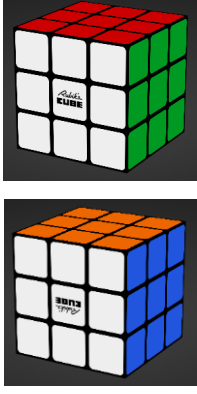
S_{B0}	<p>Rotació de 90° en el sentit de les agulles del rellotge a través de l'eix vertical.</p> <p>Estats possibles:4</p>	
S_{B1}	<p>Reflexió de les peces del costat R amb les del costat F.</p> <p>Estats possibles:2</p>	
S_{B2}	<p>Rotació de 180° en el sentit de les agulles del rellotge a través de l'eix frontal.</p> <p>Estats possibles:2</p>	

Figura 55. Taula de simetries bàsiques

CERCA DE ROBOTS JA EXISTENTS

Tots els robots en els quals ens vam inspirar podem agrupar-los en tres tipus:

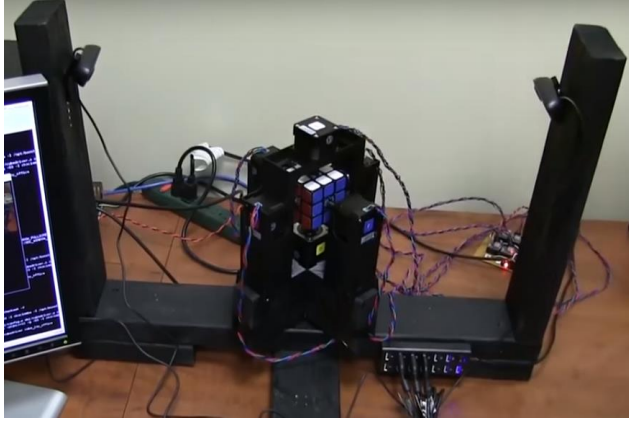
	Característica a destacar: sistema d'unió del centre del cub amb el motor.	
	AVANTATGES	INCONVENIENTS
	Rapidesa. Permet que el cub mantingui una posició fixa.	Estructura complexa.

Figura 56. Taula comparativa del tipus 1 de robot

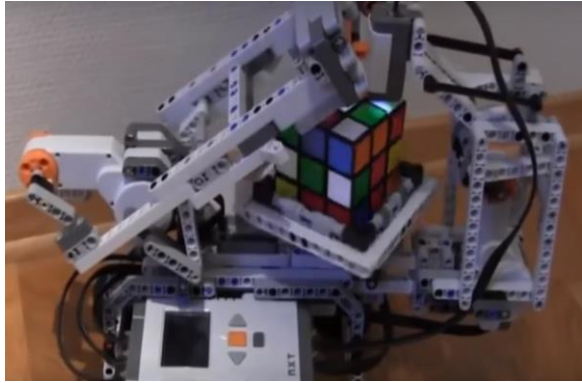
	Característica a destacar: Únicament gira la base i el cub va canviant les seves orientacions.	
	AVANTATGES	INCONVENIENTS
	Estructura més simple. Un simple motor.	Lentitud.

Figura 57. Taula comparativa del tipus 2 de robot

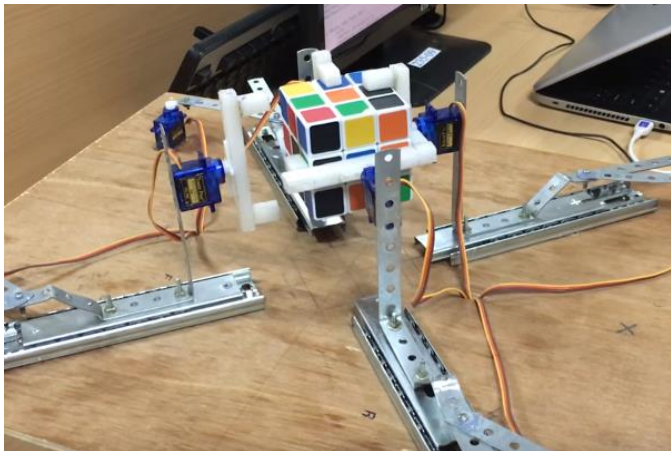
	Característica a destacar: sistema de pinça per tal de girar cada cara del cub.	
	AVANTATGES	INCONVENIENTS
	Estructura més simple. Permet que el cub mantingui una posició fixa.	Lentitud.

Figura 58. Taula comparativa del tipus 3 de robot

PROTOTIP

Prototip realitzat amb AutoCAD 2018: amb groc està marcada l'estructura, amb gris els eixos i motors, els quals estan units al cub (de color vermell)

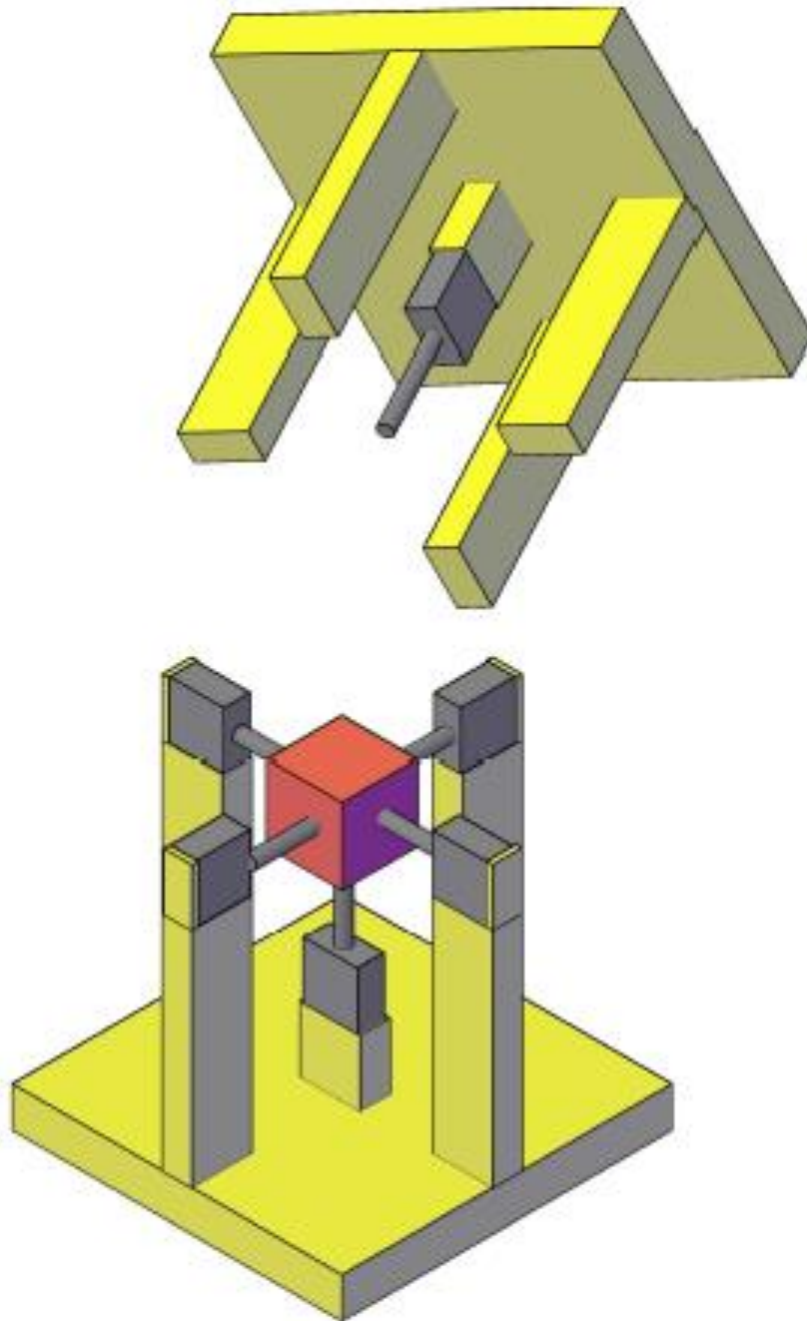


Figura 59. Prototip del robot

