

QUIMIDROID

12:45

# QUIMIDROID,

la química a la butxaca.

**Desenvolupament d'una  
aplicació per a dispositius  
Android que formuli i  
anomeni compostos  
inorgànics.**



**Departament:** Ciència i Tecnologia  
**Institut:** INS Antoni Torroja  
**Curs:** 2012-2013

**Disseny de la portada:** autors del treball.

Desembre de 2012



<b>Introducció</b> .....	5
Agraïments.....	10
<b>Objectius</b> .....	11
Estructuració i mètode de treball.....	13
<b>Fase preliminar</b> .....	15
Tria de compostos.....	17
Plataformes de programació.....	18
Selecció del nom i del logotip.....	20
<b>Desenvolupament de l'aplicació</b> .....	21
Interfície gràfica d'usuari (GUI) .....	23
Estudi d'optimització dels elements a la pantalla.....	24
Estructura i creació de pantalles.....	25
Motor de l'aplicació.....	27
Estudi del BlocksEditor.....	27
Programació i optimització dels blocs.....	29
Interpretació del codi.....	32
<b>Comprovació del funcionament</b> .....	35
Proves amb l'emulador.....	37
Cerca d'errors i solució.....	38
<b>Implantació en dispositius mòbils</b> .....	39
Compatibilitat.....	41
Empaquetatge i descàrrega.....	41
Instal·lació.....	42
<b>Conclusió</b> .....	45
<b>Recull bibliogràfic</b> .....	49
<b>Annexos</b> .....	53



# INTRODUCCIÓ





Usuari anònim: “Alguna aplicació d’Android per a la formulació inorgànica i/o orgànica?”

Respostes: “No hi ha respostes útils.”

Text 1: Pregunta trobada en un fòrum on-line.

Fig. 1. Logotip Android.

Els nostres professors de Tecnologia ens han insistit amb freqüència en què una definició adequada d’aquesta matèria era que es tractava d’una disciplina l’objectiu de la qual és satisfer necessitats humanes. Vet aquí, doncs, el motiu d’aquest treball: la bibliografia consultada, els cercadors d’Internet i altres fonts de recerca no mostren cap senyal de disposar d’una aplicació per Android que formuli compostos químics. Aquest fet ens ha motivat, doncs, a desenvolupar-ne una per aquells usuaris potencials que necessitin aquest servei.

La tria del sistema operatiu Android també la podem justificar amb facilitat. Per una banda, una altra de les característiques de la Tecnologia la constitueix la seva dimensió comercial i econòmica, no cal oblidar-ho. En aquest sentit, hem cregut assenyat – i esperem que encertat – dissenyar una aplicació química en una plataforma àmpliament distribuïda a nivell mundial pel que fa a les seves versions (vegeu la Fig. 2. on hi ha el diagrama de sectors) i, de retruc, inclosa en un dels dispositius de telecomunicacions més estès: els smartphones (vegeu la Fig. 3. on trobem la gràfica de l’evolució comercial) i/o, en general, qualsevol telèfon mòbil.

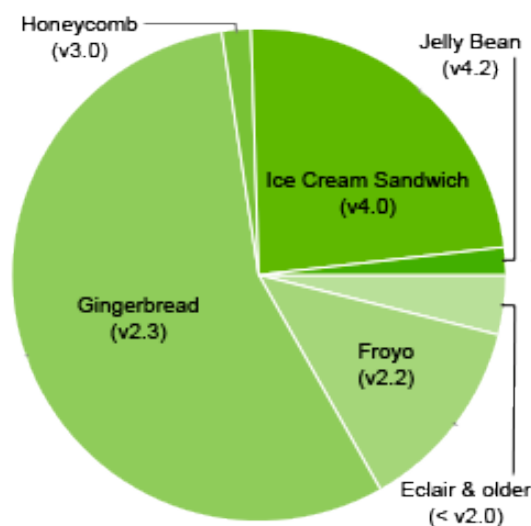


Fig. 2. Diagrama de sectors de versions d’Android.



Per una altra, existeix una plataforma de programació per a Android que presenta una filosofia molt atractiva per al programador, treballa amb un sistema de blocs fàcils de visualitzar i amb la característica original que es van acoblant, com a peces d'un puzzle. Això permet, només en un cop d'ull ràpid, copsar l'estructura general del programa.

Pel que fa a l'espectre d'usuaris que podrien trobar prou atractiva la nostra aplicació com per adquirir-la, pensem que estaria constituït per segments de població com ara estudiants de graus amb continguts de química, professionals del sector farmacèutic, químic o de la salut i, a nivell de col·lectius, departaments de serveis públics com ara el d'ensenyament, que podria incorporar-lo als seus centres com a eina pedagògica en l'aprenentatge de la química; sense oblidar els centres privats o concertats.

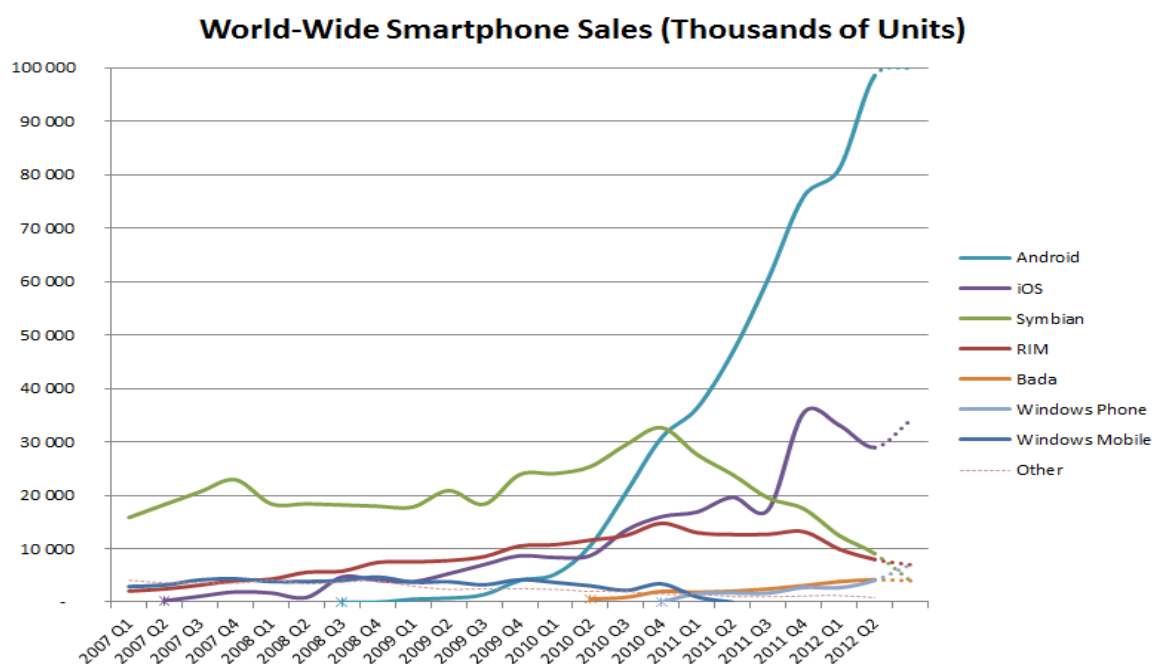


Fig. 3. Gràfica que mostra l'evolució comercial en el temps de les vendes de smartphones segons el sistema operatiu que duen incorporat.

Concretant més, el programa que us presentem formula correctament famílies de compostos inorgànics – en absència de carboni – com ara els òxids, els peròxids, els hidrurs, els àcids hidràcids, les sals binàries i els oxoàcids, a més d'oferir la possibilitat de calcular masses moleculars de molècules de fins a quatre elements. Tant els compostos com el càlcul de masses es presenten en llengua catalana i incorporen les regles d'accentuació dictades per la normativa actual.





Finalment, tan important és posar de manifest allò que es desenvolupa en el treball, com allò altre que no s'ha dut a terme, ja sigui per poca disponibilitat temporal o perquè simplement els autors del treball han considerat que s'ha assolit amb èxit l'objectiu principal que s'havien imposat, en aquest cas, demostrar que era possible dissenyar l'aplicació que us presentem. Per això no s'ha anat més enllà ni dels oxoàcids, ni ens hem endinsat en el terreny de la formulació orgànica, que hauria requerit una filosofia de programació radicalment diferent.

Tampoc no hem endegat cap intent d'estudi de mercat, que ens aproximés al volum del nínxol diferenciat de les altres aplicacions comercials preexistents, i que hauria estat complementari a la part tècnica desenvolupada. Tot aquest món per explorar el cedim als futurs entusiastes de la programació, de la química i de l'economia que vulguin continuar adobant el terreny que nosaltres només hem començat a llaurar.



## **AGRAÏMENTS:**

El desenvolupament i realització d'aquest projecte ha estat possible gràcies a la col·laboració, suport, i esforç d'un col·lectiu de persones properes a nosaltres que han cregut en ell i s'hi ha involucrant oferint-nos la seva ajuda.

Primerament, voldríem fer un agraïment al nostre tutor i exprofessor de química, Ramon Pollina i Tarrés, qui, tot i quedar sorprès davant la nostra inusual proposta de treball, ens ha donat la seva confiança acceptant portar-lo i ens ha proporcionat una ajuda molt important en la part química i ens ha aconsellat en l'acotació i la tria dels aspectes necessaris.

En segon lloc, donem les gràcies a un seguit de persones a qui hem recorregut en diverses ocasions i ens han solucionat més d'un problema: Mariona Rodríguez i Vilalta, professora de matemàtiques a l'institut Antoni Torroja que ens ha ajudat molt en el tema de la programació i va ser la primera persona que realment ens va donar crèdit en la viabilitat informàtica del projecte. A Roger Espada i Santana, estudiant d'Enginyeria Química a la Universitat de Barcelona, per l'assessorament facilitat a l'hora d'investigar les necessitats dels estudiants de grau. A Josep Dalmases i Solé, per haver-nos facilitat dispositius d'última generació per comprovar-hi el correcte funcionament de l'aplicació i extreure'n captures de pantalla. Sense tots ells, aquestes línies no haurien pogut ser escrites.

Finalment, voldríem reconèixer el suport de les nostres famílies per la seva confiança i interès constant.

**OBJECTIUS**





El present treball neix arran de la revolució que ha suposat Android en el camp dels sistemes operatius per a dispositius mòbils. Ha estat pensat i desenvolupat amb un únic objectiu: crear una aplicació que formuli compostos inorgànics per a dispositius Android. Aquesta idea, pionera avui en dia, ha constituït l'eix bàsic del treball. Aprofundint més, es tracta d'aconseguir programar una aplicació que formuli diversos compostos per satisfer una necessitat que és formular de manera ràpida i inequívoca, optimitzant així el temps emprat.

D'aquest objectiu se'n desprèn una nova motivació, divulgar la formulació química entre els estudiants prenent també l'ànim de convertir-se en una aplicació educativa i en un futur, esperem que no molt llunyà, esdevenir una eina de treball habitual com a mètode d'autocorrecció per a joves químics/es que presenti un format pràctic, portàtil i adaptat a les noves tecnologies.

## **ESTRUCTURACIÓ I MÈTODE DE TREBALL**

Un aspecte que considerem essencial i imprescindible per a poder realitzar un projecte d'aquesta magnitud amb èxit és l'ordre. Havent concretat doncs els objectius i basant-nos en ells, el primer que hem fet per tal de mantenir-lo ha sigut escriure un esborrany i marcar-nos unes pautes seguint un mètode de treball concret. Així doncs hem dividit el treball en dues etapes ben diferenciades: la programació i l'escriptura.

La primera és la que ens ha ocupat la major part del temps degut a la seva complexitat i la incapacitat de preveure els possibles entrebancs que ens hem anat trobant pel camí. En ella hem dut a terme un mètode de treball individual ja que la plataforma de programació així ens ho ha permès. Amb això hem aconseguit realitzar la feina d'una manera més eficient car cadascun dels membres podia treballar sense dependre de la disponibilitat de l'altre. Perquè aquest mètode sigui possible ens ha estat indispensable fer ús de més d'un servei en línia, com són: un xat per a comunicar-nos entre nosaltres (*Facebook*), un correu (*Gmail*) i un servei d'emmagatzematge gratuït (*Dropbox*).

La segona etapa ha estat bastant més breu i senzilla de dur a terme puix ha consistit en l'explicació de tota la feina feta fins aleshores adjuntant-hi fotografies, esquemes i taules, seguint unes pautes de disseny. L'únic canvi que hem introduït ha estat en el mètode de treball, on en lloc de fer-ho separatament ens hem reunit periòdicament.

En el fons, aquest treball és un projecte i com a tal segueix l'estructura que els caracteritza, seguidament us en mostrem un esquema adaptat:



Fig. 4. Esquema estructural que segueix el model general d'un projecte tecnològic adaptat al nostre treball.

Com es veu en l'esquema hem seguit aquesta estructura repetint-la fins a aconseguir una avaluació positiva del resultat en la qual la nostra idea quedi satisfeta.

# **FASE PRELIMINAR**







La fase preliminar en aquest treball, consisteix en acotar el treball i definir el camp de la formulació al qual ens dedicarem i és prèvia al procés de programació i a la redacció del dossier. En aquest sentit, es fa necessari seleccionar tres aspectes clau: els compostos que inclourà l'aplicació, la plataforma de programació que emprarem i el nom que tindrà. Com veiem, es tracta d'un apartat orientat a donar respostes a les preguntes inicials i que sense elles no podem començar el projecte.

## **TRIA DE COMPOSTOS**

La tria de compostos és la primera qüestió a solucionar, ja que és necessari definir fins on volem i ens és possible arribar. Per fer-la hem seguit dues premisses, en primer lloc, tenir coneixement sobre els tipus havent-los treballat prèviament i per tant, saber-los formular i anomenar amb fluïdesa; i en segon lloc, estudiar-ne la viabilitat des d'un punt de vista informàtic, temporal i dels coneixements dels quals disposem.

L'aplicació doncs està centrada en tan sols uns determinats grups de compostos binaris i ternaris, i pretén abastar la totalitat de les formes possibles dins de cadascun d'aquests. Així doncs hem decidit seleccionar els següents:

<b>TIPUS DE COMPOST I FORMA GENERAL*</b>		
Hidrurs	Hidrogen [H] + Element [E] $\longrightarrow$	$\text{EH}_x$
Òxids	Oxigen [O] + Element [E] $\longrightarrow$	$\text{E}_2\text{O}_x$
Peròxids	Òxid + Oxigen [O] $\longrightarrow$	$\text{E}_x\text{O}_{y+1}$
Sals binàries	No-metall [N] + Metall [M] $\longrightarrow$	$\text{M}_x\text{N}_y$
Hidròxids	Element [E] + ió hidroxil [OH] $\longrightarrow$	$\text{E}(\text{OH})_x$
Oxoàcids	Òxid + $\text{H}_2\text{O}$ $\longrightarrow$	$\text{H}_x\text{E}_y\text{O}_z$

Taula 1. Mètodes de formulació dels compostos inorgànics seleccionats. (\*Vegeu els annexos)



Un cop formulat el compost desitjat per l'usuari hem introduït una nova funció que ofereixi el nom de la molècula seguint estrictament la nomenclatura Stock, ja que és la recomanada per la IUPAC.

Paral·lelament també vam decidir afegir-hi una opció que permetés calcular la massa de qualsevol molècula que presenti entre un i quatre elements diferents.

## **PLATAFORMES DE PROGRAMACIÓ**

La plataforma de programació és el sistema utilitzat per realitzar l'aplicació. Cadascuna n'utilitza un determinat llenguatge de programació que és el codi amb el qual està escrita l'aplicació. Una bona selecció en aquest camp és clau ja que segons la seva complexitat, les possibilitats que ofereix i el temps disponible, una plataforma s'adequarà més a les nostres necessitats i capacitats que una altra. Android ofereix als programadors diferents possibilitats a escollir que citarem breument a continuació:

- **Eclipse:** És un entorn de desenvolupament integrat, de codi obert i multiplataforma. Requereix un bon coneixement de Java i tot el treball es realitza mitjançant línies de codi. Demana la instal·lació del Kit d'eines de Desenvolupament de Java (JDK), el Kit de Desenvolupament per a Android (SDK) i el programa en si. És totalment gratuït.
- **Basic4Android:** Aquesta plataforma utilitza el llenguatge de programació VisualBasic, similar al de Microsoft, i presenta una interfície més gràfica i menys abstracta que el Java. Es tracta de programari de pagament (entre 49\$ i 249\$).
- **Mono:** Utilitza els coneguts llenguatges de Microsoft, C# i .NET, oposats a Java. Va ser creat per una empresa anomenada "Xamarin" i fa servir el VisualStudio com a entorn de treball. L'únic que requereix és la instal·lació del SDK i la corresponent versió de Mono. Igual que l'anterior, es tracta d'una plataforma de pagament (399\$).



- **LiveCode:** Software multi plataforma, és a dir, que permet desenvolupar aplicacions per a diferents sistemes operatius. El llenguatge que fa servir s'anomena "Programació Orientada a Esdeveniments". Com la majoria, ens trobem davant d'un entorn de pagament (99\$).
- **AppInventor:** Es tracta d'una plataforma web creada per Google, que ofereix als programadors la possibilitat de programar íntegrament de manera visual fent servir un sistema de blocs molt similar al del Scratch, un software de programació visual per a Windows. Actualment es troba allotjada als servidors del MIT (Massachusetts Institute of Technology) i compta amb actualitzacions periòdiques. Presenta dues parts, l'entorn de disseny gràfic i el BlocksEditor. El seu ús és totalment gratuït.
- **InDesign CS6:** És un producte comercialitzat per l'empresa Adobe que permet crear aplicacions per a Android sense escriure una sola línia de codi. Requereix l'adquisició del programa (24,59€/mes) i la instal·lació de l'Adobe Content View.

Finalment, després d'analitzar totes les possibilitats disponibles, hem decidit utilitzar la plataforma AppInventor degut a la seva senzillesa, rapidesa a l'hora de programar i el cost nul d'ús. La seva pàgina web és [www.appinventor.mit.edu](http://www.appinventor.mit.edu) i per entrar-hi és necessària la creació d'un compte de Google.

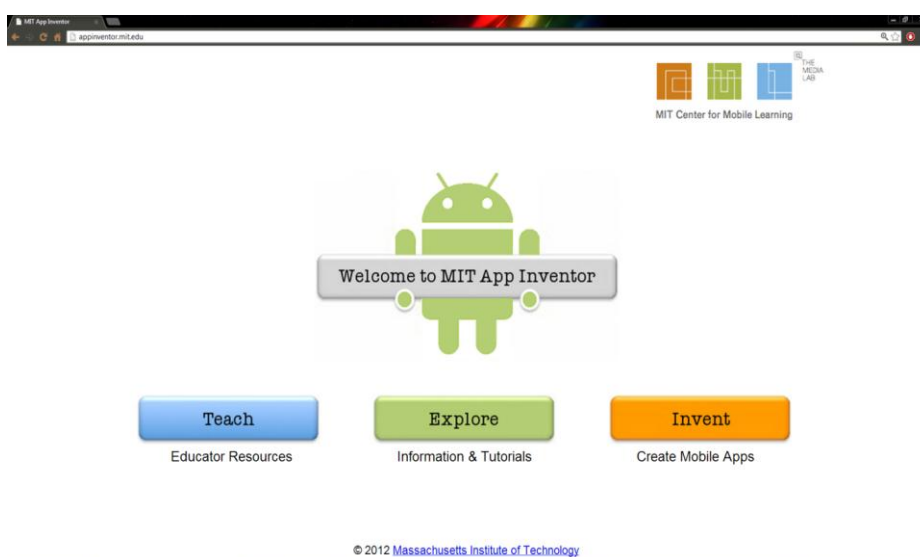


Fig. 5. Captura de pantalla de la pàgina inicial de l'AppInventor.



## **SELECCIÓ DEL NOM I DEL LOGOTIP**

Un cop hem disposat de la llista de compostos que programarem i de la plataforma a utilitzar, hem procedit a escollir el nom sota el qual la presentarem. A primera vista sembla un procés ben senzill, però hem de dir que no és així en absolut. De fet ens hem passat mesos amb diferents idees. Finalment ens hem decantat per triar el nom de “*Quimidroid*”. Com a curiositat, aquest prové de la unió de les paraules Química i Android que, com es pot apreciar, mantenen una evident relació amb el treball.

Fent referència al logotip, hem optat per la senzillesa. Ens hem decantat per una icona rodona degut al símil amb les partícules subatòmiques com ara protons o electrons, combinada amb un fons del verd que caracteritza Android i les inicials Q i D al centre. Aquest mateix disseny serveix d'icona al menú del dispositiu.



*Fig. 6 . Icona on podem apreciar les inicials Q i D.*

# **DESENVOLUPAMENT DE L'APLICACIÓ**





Un cop superada la fase preliminar, comencem el desenvolupament de l'aplicació. En aquesta etapa, possiblement la més extensa i complicada de totes però alhora la més satisfactòria per a nosaltres, hem treballat en dos camps. Per una banda, hem dissenyat la part gràfica, que és la que interactuarà amb l'usuari; i per l'altra, el motor de l'aplicació, que és l'encarregat de complir amb les funcions del software.

## INTERFÍCIE GRÀFICA D'USUARI (GUI)

Dins l'AppInventor l'apartat de disseny gràfic és on el programador crea la pantalla i és coneix com a Interfície Gràfica d'Usuari (a partir d'ara GUI). Aquest es troba dividit en diverses parts segons la funció de cadascuna: la barra de control (*Task Bar*), la paleta (*Palette*), el visualitzador (*Viewer*), el quadre de components (*Components*), el de propietats (*Properties*) i el de multimèdia (*Media*).

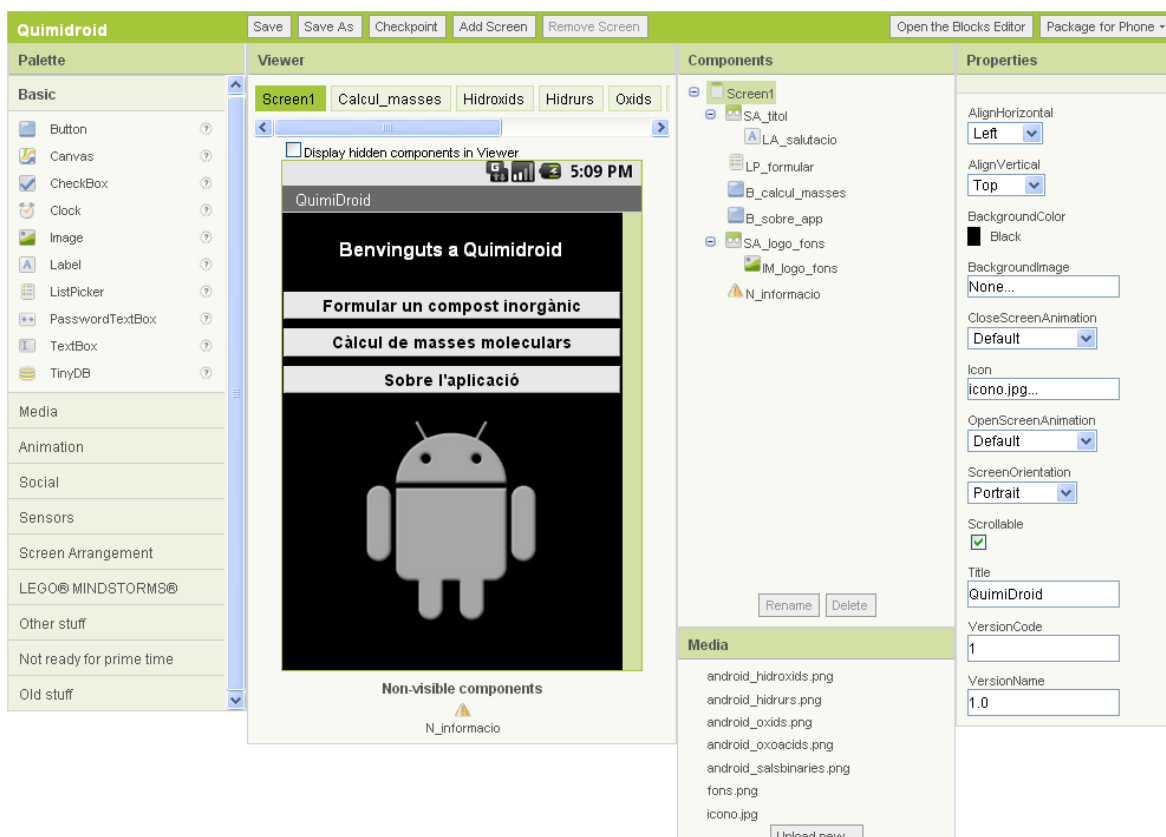


Fig. 7. Captura de pantalla de l'apartat de disseny de la pantalla.



## **Estudi d'optimització dels elements a la pantalla**

Prèviament a la col·locació dels diferents elements a la pantalla, hem realitzat un estudi basant-nos en un seguit de criteris amb l'objectiu de fer-la el més amena, atractiva i simple possible per a l'usuari. Per aconseguir-ho hem hagut de decidir diverses qüestions.

- **Col·locació dels elements:** Un aspecte clau de qualsevol aplicació o programa és tenir en compte l'anatomia humana tenint present com l'usuari governa el dispositiu on s'executa el software. En el nostre cas, això ens ha afectat, principalment, en la ubicació dels botons. Aquests els hem hagut de col·locar fixant-nos en la seva freqüència d'ús i importància, en la majoria de casos, a la meitat superior de la pantalla. Ja que utilitzem el dit polze per a treballar i és on aquest arriba amb més comoditat.
- **Tipus de lletra:** El nostre objectiu en aquesta part ha sigut trobar una lletra agradable, entenedora i que disposés de tots els símbols de la llengua catalana. Hem triat la font "*Sans Serif*", que ja venia per defecte, aprofitant que complia els requisits que buscàvem.
- **Colors:** En una societat tant visual com en la que vivim, perquè un producte agradi ens ha d'entrar per la vista. Conseqüentment, una bona selecció dels colors és fonamental per captar usuaris. Basant-nos doncs en els nostres gustos, sempre de manera fonamentada i tenint en compte el tipus d'aplicació, hem escollit la gamma de grisos, partint del blanc i acabant al negre, i verds festuc. El primer perquè dóna un toc de serietat i elegància i el segon perquè és el color característic d'Android.
- **Imatges:** Per fer més atractiu l'aspecte visual hem inclòs una imatge a cada pantalla on apareix el ninot d'Android donant instruccions.



## Estructura i creació de pantalles

En el terreny de les aplicacions d'Android s'utilitza una estructura de pantalles. Això es deu, en part, al pes d'emmagatzematge de la pantalla dins l'instal·lador i a la necessitat de fer una separació clara de les diferents funcions del programari.

En el nostre cas, seguint les línies de disseny actuals, hem estructurat la nostra aplicació en vuit pantalles.

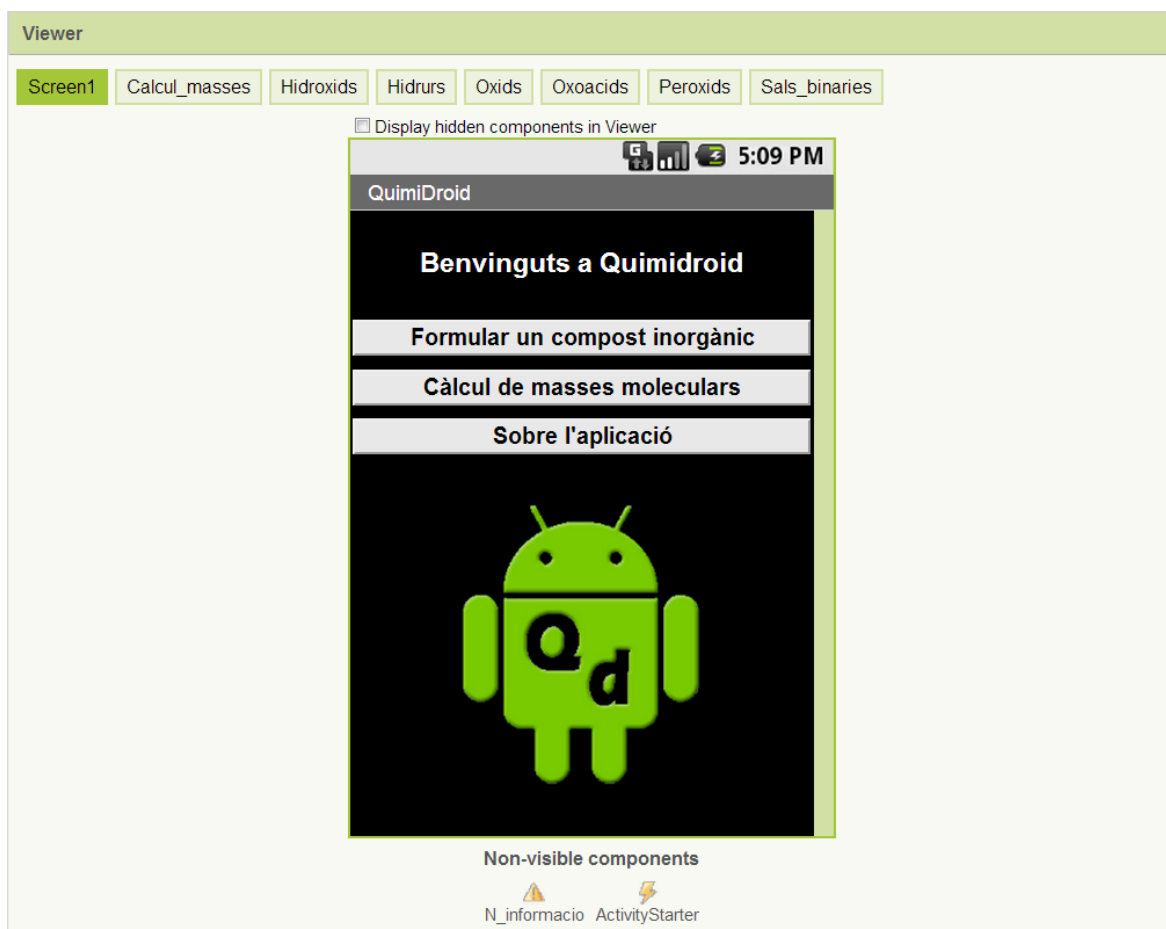


Fig. 8. Captura del "Viewer", a la part superior es mostren totes les pantalles, en primer pla veiem la Screen1.

En la imatge podem veure que cadascuna correspon a una funció diferent. D'esquerra a dreta, el menú principal, el càlcul de masses i les sis restants per als diferents tipus de compostos. A l'hora de crear-les ens demana que introduïm un nom que no contingui espais i estigui escrit únicament amb caràcters anglosaxons. Un cop creada la pantalla no és possible reanomenar-la.

Després de crear la pantalla, considerant el previ estudi d'optimització dels elements en ella, hem procedit a col·locar els diferents components que conformen l'aspecte (*ScreenArrangements*), donen informació o control a l'usuari (*imatges, botons, quadres de text, capes, etc.*) i realitzen tasques internes (*notificadors, iniciadors d'activitat, etc.*).

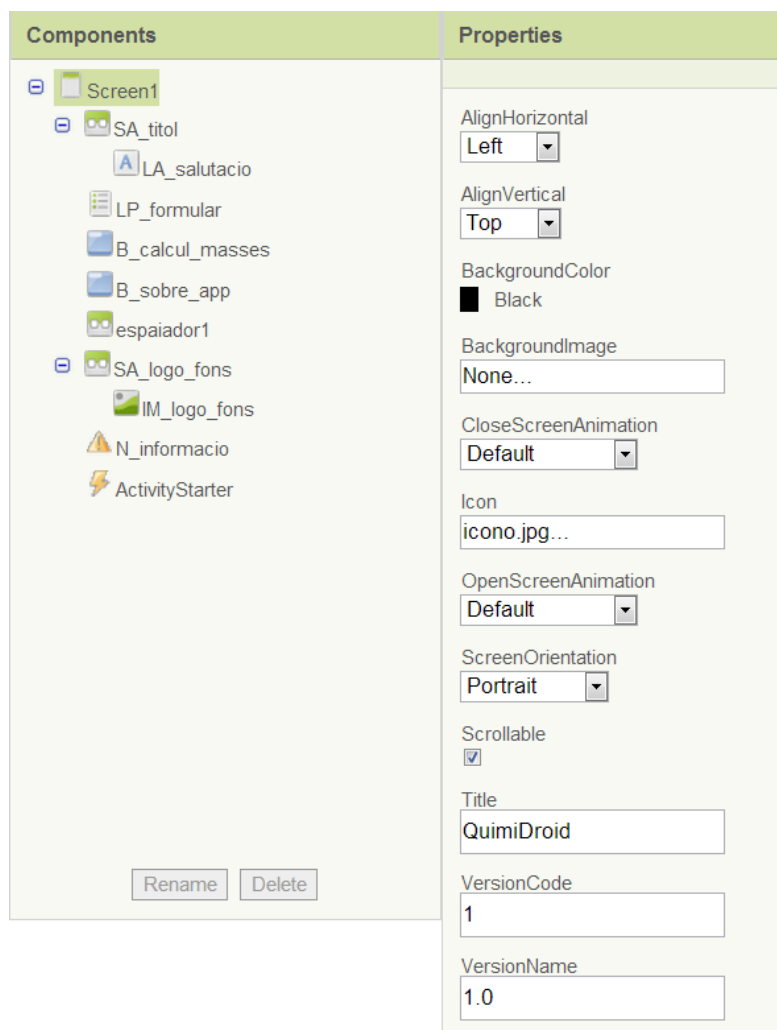


Fig. 9. Captura que mostra el quadre de components i propietats.

Com s'observa a la Fig. 9. per a facilitar-nos la feina i entendre'ns a l'hora de programar, a més de mantenir un ordre i una coherència, hem reanomenat cadascun dels components posant-los de nom les seves inicials i la funció que fan separades per una barra baixa.

## MOTOR DE L'APLICACIÓ

Fins ara hem explicat la Interfície Gràfica d'Usuari, la cara visible de l'aplicació que li confereix l'aspecte, però, si hi ha una part essencial en un programa informàtic, aquesta és, sens dubte, el seu motor. Per entendre'ns, seria l'equivalent al cervell, línies i línies de codi on trobem escrit com ha de processar les dades que l'usuari li proporciona i retornar una resposta adequada.

### Estudi del BlocksEditor

Per poder procedir a la creació del motor primer, hem hagut d'estudiar el BlocksEditor que és la màquina virtual basada en Java que hem fet servir per a programar amb blocs. Aquesta consta d'una barra de control, un espai de treball i un menú lateral.

El menú és el que conté les llibreries de blocs disponibles. En ell trobem tres pestanyes: la "Built-In", on trobem els que realitzen les funcions que no estan associades a un component de pantalla específic; la "My Blocks", on hi ha els que controlen les funcions específiques dels elements visibles a la pantalla; i finalment la

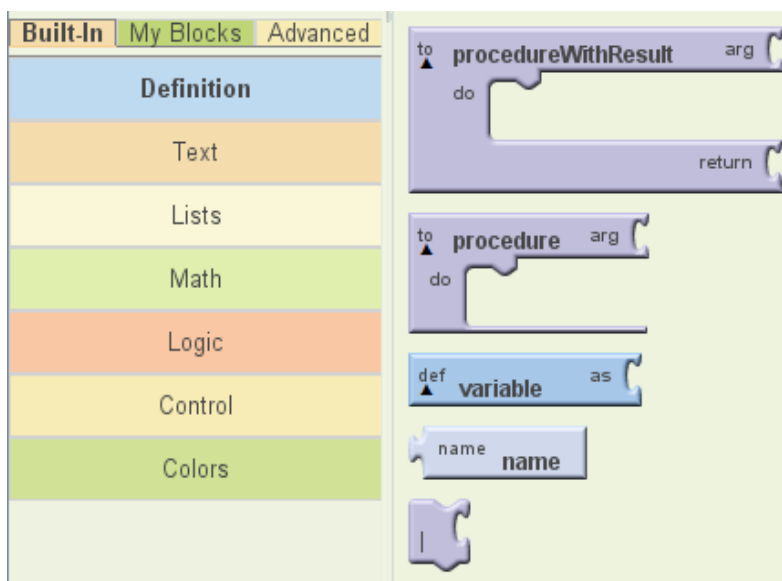


Fig. 10. Menú on hi ha les llibreries de blocs.

"Advanced" on localitzem els blocs genèrics que ens habiliten per a crear un codi més avançat, ja que són capaços de crear processos substituint algunes de les funcions predefinides de la pestanya "My Blocks".



Dins la pestanya “*Built-In*” tenim set categories segons el tipus de funció que desenvolupen els seus blocs:

- **Definitions:** En aquest cas, ens serveixen per a emmagatzemar informació en variables i crear processos.
- **Text:** Agrupa el conjunt de blocs destinats a crear i modificar textos que apareixeran dins d'altres components.
- **Lists:** Recull aquells que ens permeten treballar amb llistes, ja sigui creant, llegint o bé modificant-les.
- **Math:** Engloba els blocs que operen les funcions matemàtiques bàsiques: sumes, restes, multiplicacions, arrels, comparacions, etc.
- **Logic:** Inclou els que ens permeten treballar de forma binària, és a dir, amb dues opcions, veritat o fals. Per entendre-ho millor, podem comparar la funció d'aquest grup amb la de les portes lògiques en electrònica.
- **Control:** Com el seu nom indica, els blocs que ofereix aquesta categoria estan destinats a controlar operacions en funció d'una condició i a cridar altres processos predefinits.
- **Colors:** L'objectiu d'aquests blocs és determinar el color de certs elements.

La combinació dels blocs anteriors amb els de la pestanya “My Blocks” ens permet relacionar la pantalla amb el motor, és a dir, establir un seguit d'ordres que s'efectuen quan l'usuari interactua amb el component situat a la pantalla.



## Programació i optimització dels blocs

Un cop explicades les categories disponibles, farem referència a la manera en com es combinen els diferents blocs. Acte seguit ens ve al cap la següent pregunta: Què és un bloc? La resposta és senzilla: per entendre-ho, el podem comparar amb una peça d'un puzzle, que porta inscrit un fragment de codi en el seu interior.

Per si sol, un bloc, és una part inconnexa sense sentit ni utilitat, és a dir, no és capaç de realitzar cap funció dins el programa. Per això és necessari combinar-los entre ells per donar lloc a un codi coherent i capaç de realitzar una acció. Aquest conjunt és el que anomenem procés.

Alhora, els processos també necessiten combinar-se entre si per poder realitzar operacions més complexes. Alhora aquests s'agrupen en el que coneixem com a pantalla. Per explicar-ho d'una manera més comprensible, ho podem comparar amb una cadena de muntatge d'una fàbrica, on els treballadors serien equivalents als processos i la cadena, mirada com una unitat, correspondria a una pantalla.

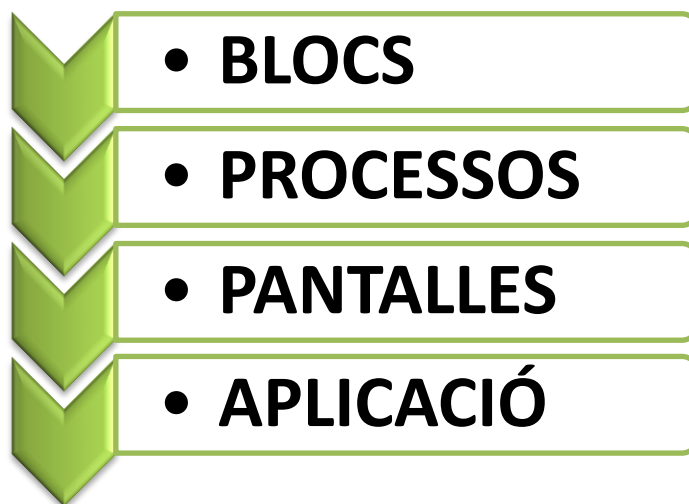


Fig. 11. Esquema jeràrquic que mostra l'ordre d'unió dels elements.

Així doncs tenim un conglomerat on cadascun dels treballadors, els processos, tenen adjudicada una funció concreta i un medi, la pantalla, on es desenvolupen aquestes. L'execució del conjunt d'operacions dóna lloc a un producte final, que en el nostre cas és la resposta a la petició feta per l'usuari, ja sigui un compost o una massa molecular.



Finalment, trobem que una o més pantalles unides entre elles formen l'aplicació. Fent, un cop més, un símil amb la fàbrica, l'agrupació de cadenes de muntatge, les pantalles, dóna lloc a la fàbrica en si, l'aplicació.

Veiem doncs, com podem apreciar a la Fig. 12., que les aplicacions presenten una forma piramidal que es basa en agrupacions per configurar els nous nivells.

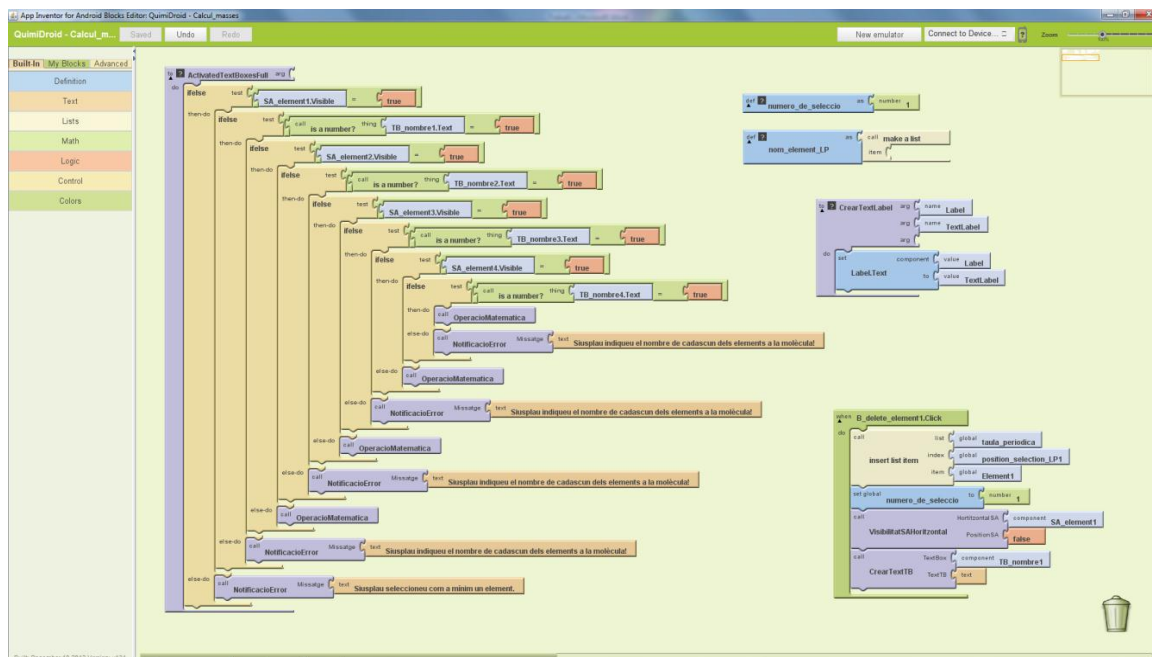


Fig. 12. Conjunts de blocs que conformen el motor i contenen el codi.

Explicada ja l'estructura interna del motor, hem de posar atenció en un altre aspecte important, la informació amb la qual treballarà aquest. La matèria primera per a formular compostos són elements de la taula periòdica i les seves valències, en conseqüència vam haver d'introduir aquesta informació dins el codi.

La nostra intenció inicial era crear la base de dades necessària mitjançant taules en full de càlcul, és a dir, fent ús d'un programa extern. Ara bé, en el moment que vam començar a fer proves per pujar el document i a buscar informació sobre com fer-ho es va fer patent la impossibilitat d'utilitzar aquest mètode ja que l'AppInventor no ho permet.

Per solucionar el problema hem creat les taules necessàries definint variables amb llargues llistes, ordenat els elements segons el període al qual pertanyen i les valències que presenten; i hem creat complexos processos que les relacionessin entre elles.



Superats tots els entrebancs i un cop hem aconseguit programar tot el motor de l'aplicació, tot i que encara presentava alguns errors, permetia formular i anomenar de forma correcta gairebé tots els compostos excepte en algun cas. Tanmateix, el codi que teníem era poc polit, ja que era massa llarg; i poc eficient en referència als temps de resposta. A fi de millorar aquests dos aspectes hem procedit a l'optimització.

Aquesta fase és similar a l'estudi d'optimització dels elements a la pantalla que hem fet abans, amb la diferència que aquest cop enlloc de treballar amb els components de la GUI ens hem ocupat de la millora del codi. La pregunta que ens ve al cap automàticament és: i això com ho farem? Bé, pot semblar difícil de resoldre, però no, en realitat és bastant lògic i no massa complicat: evitant les repeticions i dividint la feina en processos.

Un dels principis fonamentals de la programació és el que ens marca el camí a seguir a l'hora d'optimitzar el segon aspecte citat. Aquest ens diu que la millor aplicació és aquella que realitza la funció desitjada amb el menor nombre de línies de codi possibles. Seguint això, el primer aspecte, l'hem solucionat creant un procés genèric per a cada tipus d'operació. D'aquesta manera enlloc de tenir l'acció escrita nombroses vegades, només ha estat necessari programar-la un cop i cada vegada que ha estat requerida hem cridat el procés i li hem definit els camps d'actuació. Amb això hem retallat la llargada del codi i a la vegada els temps de resposta.

Com va dir Juli Cèsar: *"Divideix i venceràs"* i el nostre cas no n'és pas una excepció. Fent-li cas, per complir el segon aspecte, hem dut a terme una divisió igualitària de les operacions entre les diferents accions de l'usuari sempre que ens ha estat possible. Un exemple per entendre-ho seria el següent: l'usuari vol formular un hidrur. Per respondre a la petició l'aplicació ha de definir el text de sis capes diferents segons el que ell seleccioni en dos *ListPickers*. Enlloc de carregar tota aquesta feina a l'últim *ListPicker* seleccionat en fem un repartiment equitatiu: el primer definirà tres textos i el segon tres més. Això tan simple que fins i tot pot semblar una estupidesa ens permet aconseguir una reducció dels temps de resposta notòria.

## INTERPRETACIÓ DEL CODI

Finalment, un cop considerem que el motor està completament construït, ens disposem a mostrar com el dispositiu l'interpreta i ens retorna el resultat de l'acció sol·licitada. Dit d'una altra manera, el què veurem en aquest apartat és una comparació entre el codi del motor i el que en resulta d'aquest com a resposta a la pantalla on l'usuari interactua. Així doncs anem a fer un anàlisi ràpid i planer de dos fragments del codi del motor.

El primer cas es tracta d'un petit procés que hem anomenat "Apostrofació" i s'encarrega de determinar si la preposició "de" del nom del compost formulat anirà apostrofat o no amb l'objectiu d'aconseguir la màxima correcció lingüística possible. Un exemple d'aquest cas és *l'òxid d'estronci* o *l'hidrur d'antimoni*.

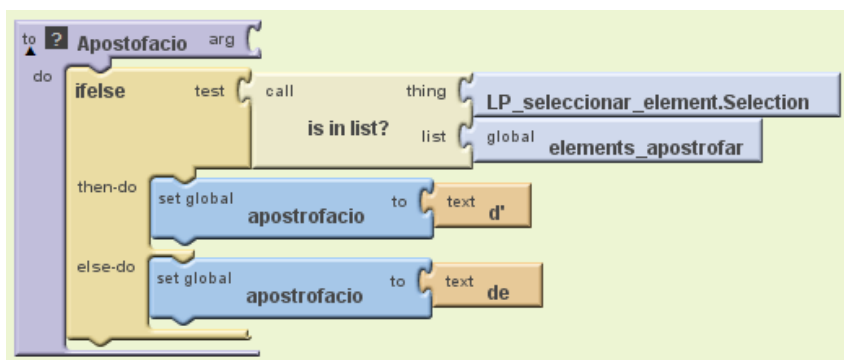


Fig. 13. Codi del procés que determina si la preposició "de" s'apostrofa o no.

Fixant-nos en les imatges anteriors veiem com el procés s'activa alhora d'anomenar la molècula seleccionada per l'usuari i com treballa. Primer veiem una condició "ifelse" que comprova si els blocs encaixats en la ranura "test" coincideixen amb la comprovació. Si el resultat és afirmatiu s'executa el codi de la secció "then-do" i en cas contrari els del "else-do". La funció que fan els que es troben en aquests apartats és donar un valor a la variable "apostrofació" que emmagatzemarà el text "d" o bé "de" segons l'element seleccionat.

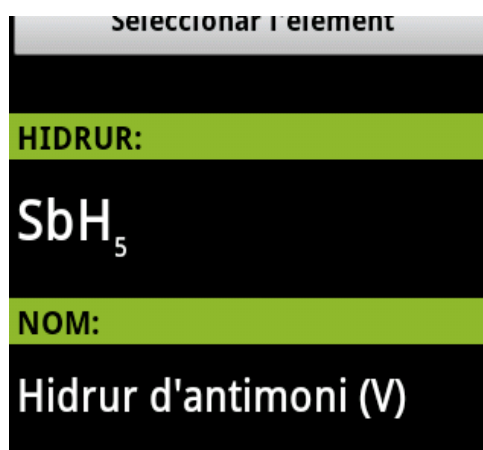


Fig. 14. Retall d'una captura de pantalla on es mostra l'acció del procés anterior.



En el segon i últim cas tenim un fragment de la funció del càlcul de masses que té la missió de calcular la massa molecular d'una molècula que presenti entre un i quatre elements.

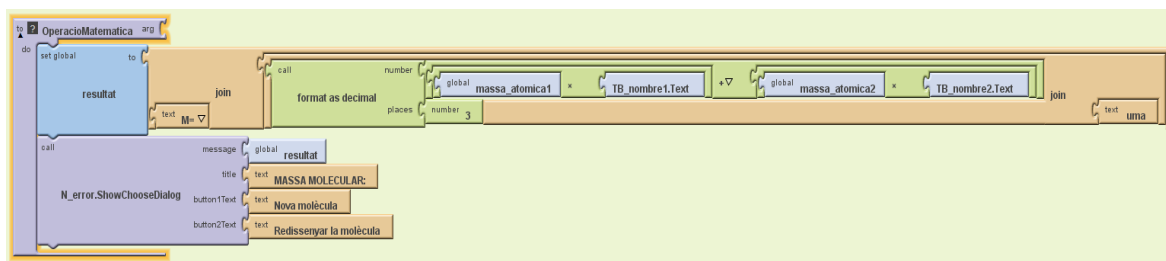


Fig. 15. Procés encarregat de calcular la massa molecular de la molècula introduïda per l'usuari



Fig. 16. Captura de pantalla on és veu el resultat de l'operació duta a terme pel procés anterior dins el notificador.

Si analitzem un cop més un altre fragment ens adonem que es tracta d'un procés encarregat de dur a terme l'operació matemàtica que dóna el valor de la massa de la molècula indicada per l'usuari. En ell veiem dos blocs consecutius que porten units altres blocs. El primer, de color blau, realitza l'emmagatzematge del resultat de l'operació en una variable anomenada "resultat" i el segon, de color violeta, s'ocupa de fer aparèixer el notificador, que podem veure a la imatge de l'esquerra. Aquesta funció es presenta com un requadre de color blau on s'hi mostra el valor demanat per l'usuari.

Així doncs havent fet una petita introducció en el què ve a ser la programació pròpiament dita, ens dirigim al proper apartat on realitzarem la comprovació del correcte funcionament d'aquest codi.



# **COMPROVACIÓ DEL FUNCIONAMENT**





La comprovació del funcionament és l'última fase de la programació abans de poder implantar l'aplicació en el dispositiu. En ella el que hem buscat ha estat la fiabilitat, és a dir, la correcció de cadascun dels problemes que hem anat detectant. Aquesta etapa es duu a terme conjuntament amb el desenvolupament del motor perquè per poder seguir programant és indispensable assegurar-se que el codi que tenim és correcte.

## **PROVES AMB L'EMULADOR**

Davant la necessitat de comprovar el bon funcionament del codi que hem anat escrivint durant el procés de programació, és imprescindible l'ús d'un emulador. La funció d'aquest és crear una màquina virtual dins l'ordinador que simuli ser un dispositiu amb sistema operatiu Android.

El programa encarregat de fer això es troba disponible juntament amb l'AppInventor sense l'obligació d'instal·lar a l'ordinador cap software addicional. Per carregar l'aplicació l'únic que cal fer és obrir-lo i des del *BlocksEditor* connectar-lo, fet que ens estalvia tot el lent procés d'empaquetatge, transferència i instal·lació en el dispositiu.

No obstant els avantatges que ens ofereix trobem en ell algunes limitacions com ara la impossibilitat de canviar de pantalla o bé provar la rotació automàtica. També cal dir que és possible fer servir com a emulador el mateix dispositiu si el connectem a través d'un USB o via WIFI a l'ordinador.



Fig. 17. Emulador d'Android per a PC.

## CERCA D'ERRORS I SOLUCIÓ

Una vegada establerta la connexió entre el simulador i el *BlocksEditor*, comencem a utilitzar l'aplicació amb l'objectiu de trobar els errors en el codi. Quan en trobem un el programa ens ho indica amb una notificació emergent i una descripció tècnica per a localitzar-lo. A continuació es mostren alguns casos (Fig. 14.):



Fig. 18. Diferents notificacions d'errors trobats amb l'emulador.

Esdevingut això, procedim a la localització exacta de l'error dins el *BlocksEditor* i analitzem el motiu pel qual falla. Després de comprendre perquè el codi és incorrecte procedim a la correcció d'aquest. Tot seguit, un cop resolt, tornem a iniciar el procés de comprovació amb el simulador que només acaba quan no es detecta cap nou error.

La majoria d'errors descoberts són a causa de la imprevisió d'un cas concret o un determinat aspecte, la confusió de blocs o bé la mala unió d'aquests.

# **IMPLANTACIÓ EN DISPOSITIUS MÒBILS**







Un cop hem comprovat de forma exhaustiva el funcionament de l'aplicació en l'emulador obtenint resultats satisfactoris, és a dir, havent solucionat tots els errors coneguts i sense que en sorgeixin de nous, comença la darrera fase del treball. Aquesta última fase consisteix en passar de l'emulador al propi dispositiu mòbil. Per fer això, primer de tot, és necessari comprovar la compatibilitat dels dispositius amb l'aplicació, tot seguit realitzem el procés d'empaquetatge del codi amb la finalitat de generar l'instal·lador i finalment executem la instal·lació.

## **COMPATIBILITAT**

Per poder fer servir l'aplicació el dispositiu ha de complir un únic requisit indispensable: portar instal·lat el sistema operatiu Android. Per tant és compatible amb tablets, smartphones i ordinadors portàtils, ja que són els productes que trobem al mercat amb aquesta característica.

## **EMPAQUETATGE I DESCÀRREGA**

Una vegada hem guardat l'aplicació i estem segurs del bon funcionament procedim a empaquetar el codi per obtenir l'instal·lador. Per fer-ho fem servir l'eina d'empaquetatge que ens ofereix l'AppInventor. Aquesta ens brinda les possibilitats que es mostren en la imatge.

Nosaltres seleccionem la segona, que és la que descarrega a l'ordinador el paquet d'instal·lació, i es mostra una barra amb l'estat del progrés.

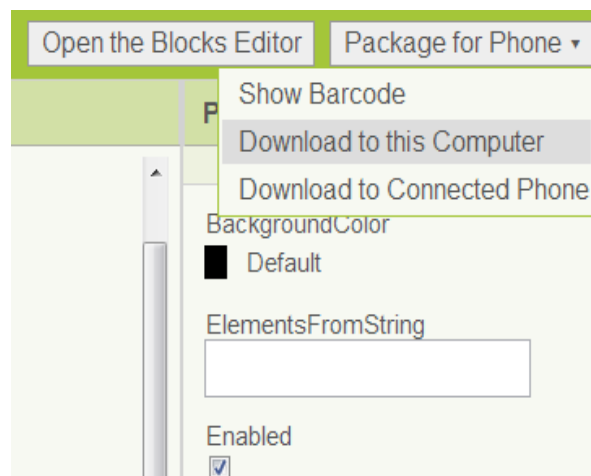


Fig. 19. Menú d'empaquetatge de l'instal·lador.

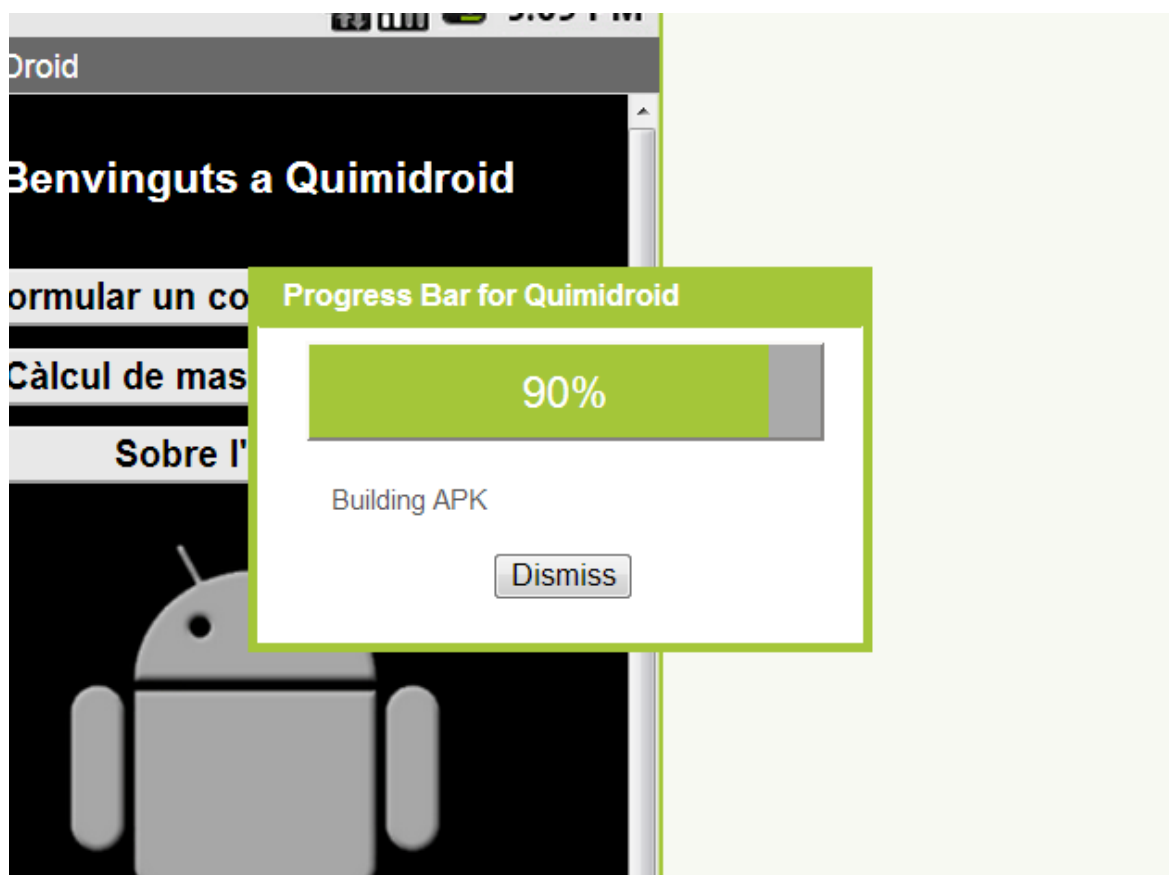


Fig. 20. Captura de pantalla de la barra del progrés d'empaquetatge.

Un cop la barra hagi arribat al 100% se'ns obre una finestra on seleccionem el directori on es guardarà l'arxiu.

## INSTAL·LACIÓ

El pas definitiu per poder utilitzar Quimidroid des d'un dispositiu mòbil és la instal·lació. En aquest pas, un cop hem empaquetat l'aplicació i disposem de l'instal·lador al nostre ordinador, connectem l'aparell, ja sigui via USB o Bluetooth, i transferim l'arxiu a la memòria d'aquest.

Seguidament ens dirigim al directori on hem guardat l'instal·lador i l'obrim, tot seguit apareix una pantalla que ens pregunta si desitgem continuar i ens avisa dels permisos necessaris per a fer-ho. Premem el botó "*Instal·lar*" i s'inicia el procés que pot tardar uns minuts i esperem a que finalitzi. Finalment cliquem el botó "*Obrir*" i ja estem llestos per començar a utilitzar-la.



Fig. 21. Fotografies del procés d'instal·lació en un smartphone Android.

Aquesta etapa ha exigut ser repetida diversos cops ja que han aparegut nous errors degut al canvi de medi on s'executa l'aplicació, que ha posat de manifest mancances que amb l'emulador no eren visibles. Un exemple d'això ha sigut el desajustament de la col·locació dels elements en la pantalla quan l'hem instal·lat en un smartphone amb una pantalla més gran.

Davant aquest imprevist ens hem vist obligats a tirar enrere en el treball i repetir l'apartat anterior per tal de solucionar els problemes i comprovar de nou el bon funcionament per poder reprendre satisfactòriament la implantació en els dispositius.



# CONCLUSIÓ





1. El treball desenvolupa una aplicació per a dispositius Android focalitzada en la formulació inorgànica basant-se en les normes de la IUPAC (International Union of Pure and Applied Chemistry) i el càlcul de masses. Aquesta aplicació, restringida només a certs tipus de compostos, ve motivada pel fet de ser una idea pionera i sense implantació en el mercat alhora que al seu potencial pedagògic.
2. D'entre tots els tipus de compostos inorgànics, com ja hem citat en la fase preliminar del treball, n'hem realitzat una tria dels que hem considerat més comuns i simples. Aquest fet ha estat basat en dos factors: el primer ha estat que el temps disponible era finit. El segon, obeeix a la dificultat de cadascun a ser programat. En aquest sentit hem estat molt afortunats d'haver trobat l'ApplInventor com a plataforma de programació.
3. Després d'aplicar el procés de desenvolupament hem aconseguit crear una versió estable de l'aplicació projectada, assolint l'objectiu proposat i cobrint tots els tipus de compostos inorgànics previstos.
4. Després d'efectuar proves amb l'aplicació instal·lada en smartphones, mitjançant la col·laboració d'estudiants, ens ha estat possible comprovar la seva utilitat. Els resultats obtinguts d'aquestes han estat molt positius tant pel que fa als temps de realització dels exercicis de formulació com a la satisfacció i valoració dels joves participants.

Full	Mètode de formulació	Nombre d'exercicis	Nombre de compostos de cada tipus	Temps emprat
1	Manual	18	3	10 min
1	Aplicació	18	3	4 min
2	Manual	24	4	13 min
2	Aplicació	24	4	6 min

Taula 2. Comparació temps de realització dels exercicis.



5. També és pertinent, tot i que resulta lògic, assenyalar que els compostos ternaris representen un major obstacle a l'hora de programar-los respecte als binaris degut al considerable augment de la llargada del codi. Així també cal dir que la dificultat de programació a la qual ens hem d'enfrontar es troba en una relació directament proporcional a l'estalvi temporal de l'usuari.
6. Pel que fa a la importància dels compostos, ens hem adonat que els més senzills són la base dels més complexos. Això significa, en el nostre cas, que els hidrurs i els òxids són la base dels altres, exceptuant les sals binàries. Aquest fet ens ha estat de gran ajuda ja que ens ha facilitat la programació perquè hem pogut anar aprofitant el codi ja escrit amb l'única necessitat d'introduir-hi petites modificacions.
7. No podem finalitzar un treball com aquest sense posar èmfasi en:
  - a. El fet que tots els compostos que permet formular Quimidroid són teòrics i malgrat seguir estrictament les regles de formulació, en cap cas podem garantir la seva existència en la natura o la possibilitat de sintetitzar-los artificialment, ja que ni hem dut a terme ni hem contemplat fer tal comprovació.
  - b. Hem de dir que les característiques tabulades i utilitzades per a cada element han estat extretes de la taula periòdica adjuntada amb el llibre "*Química 1 Batxillerat*" de l'editorial McGrawHill i per tant ens hem cenyit a la informació present en ella.
  - c. També fer referència al fet que tots els compostos són anomenats amb la nomenclatura Stock, per aquest motiu, tot i que alguns siguin més coneguts pel seu nom popular, en l'aplicació no hi figuren.
8. Finalment volem fer un petit apunt: aquest treball el donem per finalitzat aquí però el projecte va més enllà i és molt més ampli ja que pretenem introduir Quimidroid al Google Play (el mercat d'aplicacions Android) en un futur pròxim amb l'objectiu de donar-li continuïtat i seguir-la millorant perquè els usuaris se'n puguin beneficiar.



# **RECULL BIBLIOGRÀFIC**





## **WEBGRAFIA**

1. [www.beta.appinventor.mit.edu](http://www.beta.appinventor.mit.edu)
2. [www.google.cat](http://www.google.cat)
3. [www.webadicto.net](http://www.webadicto.net)
4. [www.usuarios.multimania.es](http://www.usuarios.multimania.es)
5. [www.androidinventor.es](http://www.androidinventor.es)
6. [www.appinventorblocks.com](http://www.appinventorblocks.com)
7. [www.groups.google.com](http://www.groups.google.com)
8. [www.wikipedia.org](http://www.wikipedia.org)
9. [www.androideity.com](http://www.androideity.com)
10. [www.youtube.com](http://www.youtube.com)
11. [www.elandroidlibre.com](http://www.elandroidlibre.com)
12. [www.slideshare.net](http://www.slideshare.net)
13. [www.yahoo.com](http://www.yahoo.com)
14. [www.iesromerovargas.net](http://www.iesromerovargas.net)
15. [www.iupac.org](http://www.iupac.org)
16. [www.wordreference.com](http://www.wordreference.com)

## **LLIBRES I TREBALLS**

1. CASTELLS, P.; RIBA, N. i ANDREU, F. *“Química 1 Batxillerat”*. McGraw-Hill, 1a edició, Madrid (2008).
2. COTTON, F. A. i WILKINSON, G. *“Química Inorgánica Avanzada”*. Editorial Limusa, 4a edició, Mèxic (1986).



3. RIBAS LEQUERICA, J. *“Desarrollo de aplicaciones para Android”*. Anaya Multimedia, 1a edició, (juny 2011).
4. ORIOL, E. i ROYES, A. *“Pesticides 1 – Futbolistes 0, Una anàlisi de la correlació entre la toxicitat dels fitosanitaris aplicats a la gespa dels estadis de futbol i la mort sobtada dels jugadors”*. Treball de recerca de l'INS Antoni Torroja (2011).
5. ORIOL i MARTÍ, T. *“Fabricació de tubs d'acer de precisió”* Treball de recerca de l'INS Antoni Torroja (2007).

# **ANNEXOS**





Procés de formulació per a compostos inorgànics.....	57
Dades i estadístiques d'Android.....	61
Curiositats.....	65
Entorn de l'AppInventor.....	69





# **Procés de formulació per a compostos inorgànics**





### **Hidrurs:**

1. Es col·loca l'element seleccionat de la taula periòdica a l'esquerra actuant amb la valència que haguem seleccionat i l'hidrogen a la dreta actuant sempre amb valència -1.
2. Intercanviem valències i anomenem.

\*\* En el cas dels elements dels grups XVI, XVII l'element el col·loquem a la dreta i l'hidrogen a l'esquerra, això és degut a la major electronegativitat.

### **Òxids:**

1. L'element seleccionat es posiciona a l'esquerra amb la seva valència i l'oxigen a la dreta sempre amb valència -2.
2. Intercanviem valències, si és possible simplifiquem i anomenem.

### **Peròxids:**

1. Formulem l'òxid desitjat i li sumem un àtom d'oxigen.
2. NO simplifiquem i anomenem.

### **Sals binàries:**

1. Posem el metall a l'esquerra amb la seva valència, agafem el no-metall i el posem a la dreta amb la corresponent valència.
2. Intercanviem valències, si és possible simplifiquem i anomenem.



### **Hidròxids:**

1. Agafem el grup hidroxil (OH) i el posem a la dreta sempre amb valència -1, seleccionem l'element desitjat i el posem a l'esquerra amb la seva valència.
2. Intercanviem valències i anomenem.

### **Oxoàcids:**

1. Formulem l'òxid amb l'element desitjat.
2. Li sumem una molècula d'aigua (H<sub>2</sub>O).
3. Ordenem seguint la forma H<sub>x</sub>E<sub>y</sub>O<sub>z</sub> on E representa l'element.
4. Simplifiquem si és possible i anomenem.

Recordar un cop més que tots els compostos s'anomenen seguint la nomenclatura Stock de la IUPAC.

# **Dades i estadístiques d'Android**





VERSIÓ	NOM PLATAFORMA	DATA LLANÇAMENT
1.0	-	23 de setembre de 2008
1.1	Petit Four	9 de febrer de 2009
1.5	Cupcake	30 d'abril de 2009
1.6	Donut	15 de setembre de 2009
2.0/2.1	Eclair	26 d'octubre de 2009
2.2.x	Froyo	20 de maig de 2010
2.3.x	Gingerbread	6 de desembre de 2010
3.x	Honeycomb	22 de febrer de 2011
4.0.x	Ice Cream Sandwich	19 d'octubre de 2011
4.1/4.2	Jelly Bean	9 de juliol de 2012

Taula 3. Dades de les diferents versions d'Android.

ANY	MES	APLICACIONS DISPONIBLES	DESCÀRREGUES
2009	Març	2,300	-
	Desembre	16,000	-
2010	Març	30,000	-
	Abril	38,000	-
	Agost	80,000	1.000 milions
	Octubre	100,000	-
2011	Maig	200,000	3.000 milions
	Juliol	250,000	6.000 milions
	Octubre	319,000	-
	Desembre	380,297	10.000 milions
2012	Gener	400,000	-
	Febrer	450,000	-
	Maig	500,000	-
	Juny	600,000	20.000 milions
	Novembre	700,000	25.000 milions

Taula 4. Evolució de les aplicacions al Market.





# Curiositats





### **D'on prové el nom Android?**

El nom Android fa al·lusió a la novel·la de Philip K. Dick "*¿Somien els androides en ovelles elèctriques?*", que fou posteriorment adaptada al cinema amb el nom de *Blade Runner*.

### **Perquè les diferents versions d'Android reben noms tant inusuals?**

Les versions d'Android són anomenades amb el nom de diferents postres anglesos. Cada vegada que es llança una nova versió se li dona el nom d'unes postres que comencin amb una lletra diferent seguint un ordre alfabètic.

- A. Apple Pie (v1.0), "*Pastis de poma*"
- B. Banana Bread (v1.1), "*Pa de plàtan*"
- C. Cupcake (v1.5), "*Magdalena glacejada*"
- D. Donut (v1.6), "*Rosquilla*"
- E. Éclair (v2.0/v2.1), Pastis francès conegut com "*Petisú*"
- F. Froyo (v2.2), (Abreviatura de «Frozen Yogurt») "*logurt gelat*"
- G. Gingerbread (v2.3), "*Pa de gingebre*"
- H. Honeycomb (v3.0/v3.1/v3.2), "*Panell de mel*"
- I. Ice Cream Sandwich (v4.0), "*Entrepà de gelat*"
- J. Jelly Bean (v4.1/v4.1.2/v4.2), "*Mongeta de gelatina*"



# **Entorn de l'ApplInventor**



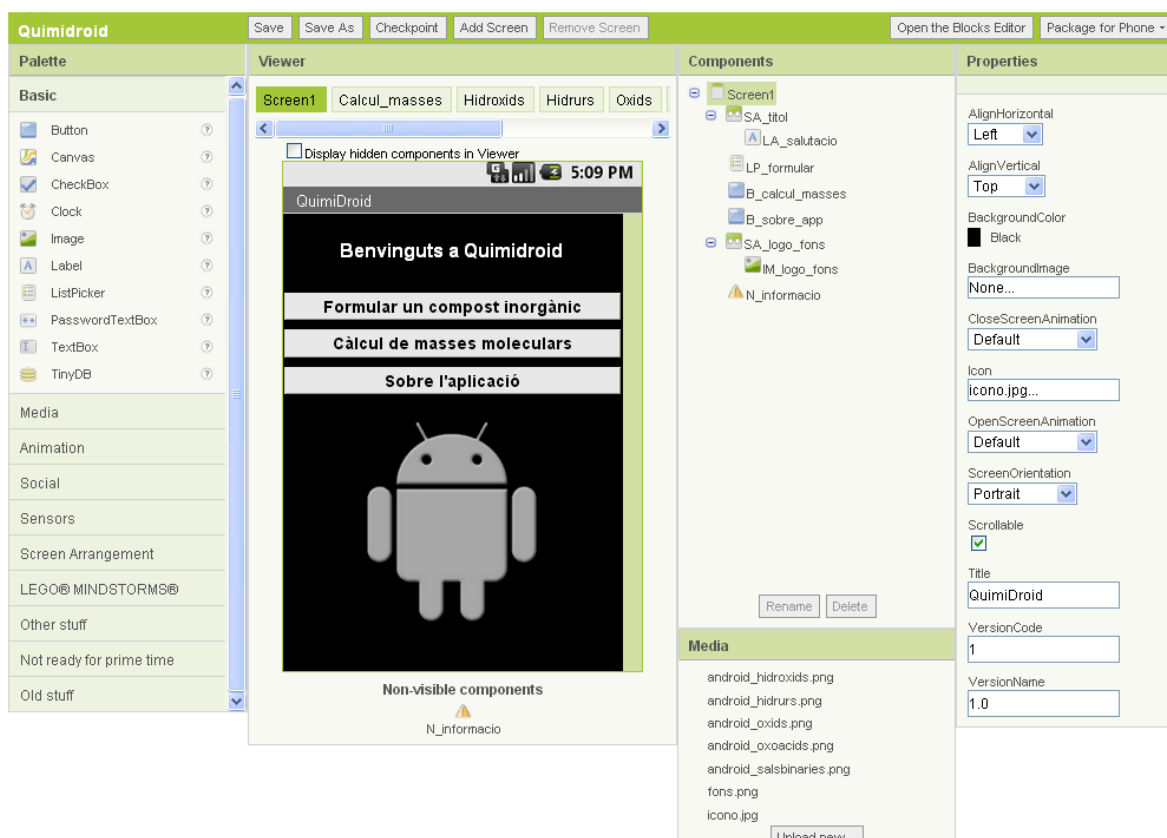


Fig. 22. Captura de pantalla de l'espai de treball de la Interfície Gràfica d'Usuari de l'AppInventor.

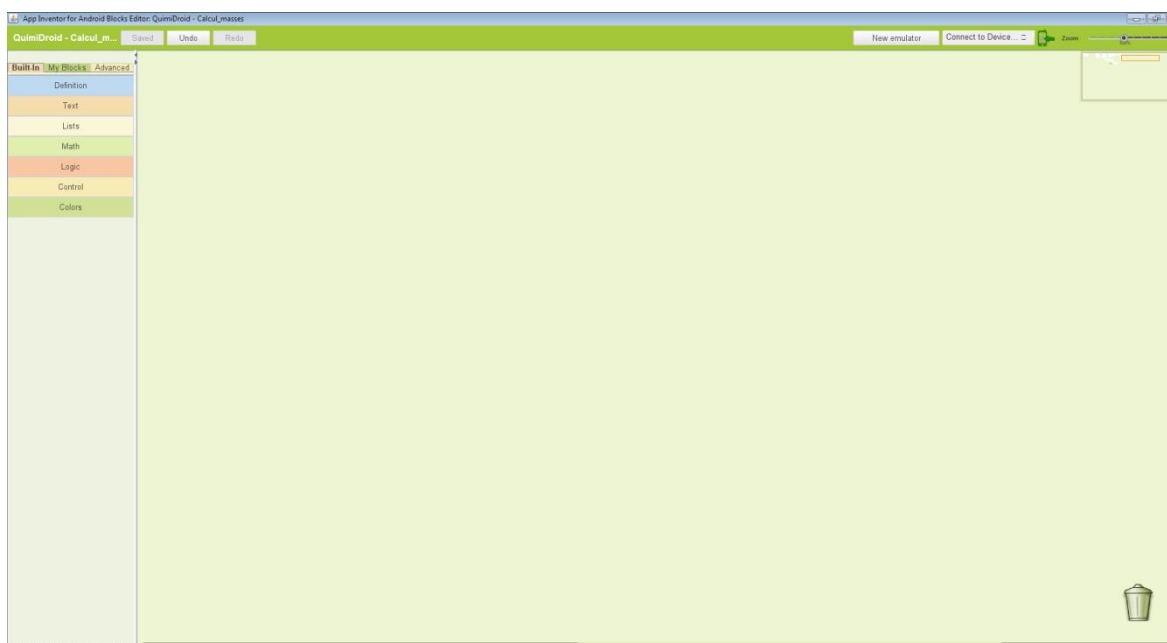


Fig. 23. Captura de pantalla de l'espai de treball del BlocksEditor.