



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques
Universitat de Barcelona**

EQUIDIA: DIAGRAMES PER A LA INTERPRETACIÓ VISUAL D'EQUACIONS EN DISPOSITIUS MÒBILS

Oriol Janés Pereira

Tutors: Sergio Escalera Guerrero
Oriol Pujol Vila

Realitzat a: Departament de Matemàtica
Aplicada i Anàlisi. UB

Barcelona, 20 de Gener de 2013

Agraïments

Al finalitzar un projecte tant llarg i ple de dificultats com és un Projecte de Fi de Grau, que marca la culminació d'aquests estudis universitaris, és inevitable sentir-se assaltat per un sentiment d'orgull sobre el treball fet. No obstant, mirant enrere des d'una perspectiva més objectiva, han sigut moltes les persones que m'han donat una empenta, m'han donat un cop de ma o m'han encarrilat quan no veia les coses gens clares. Per això vull dedicar aquest espai a agrair la seu paper en aquesta llarga etapa.

He d'agraciar primerament al Doctor Sergio Escalera Guerrero i al Doctor Oriol Pujol Vila el haver-me ofert treballar en aquest projecte. El seu suport i la seva confiança en el meu treball han sigut summament importants, així com els seus consells i el haver-me aportat els mitjans i el material necessari per haver tirat endavant aquest projecte.

Agraeixo també de manera molt significativa a Josep Tarrés i Turon, Professional independent en el sector de les publicacions, el qual ha sigut el pare del sistema que ha fet possible aquest projecte, sense el seu suport i les seves constants aportacions res d'això hagués estat possible.

Per als meus amics i companys d'universitat, Xavier Vales, Antonio Torralba i Marc Moreno, també un fort agraïment per haver suportat més carrega de treball per haver-me dedicat a aquest projecte amb tanta força i per haver-me acompanyat durant tota la carrera.

Finalment, l'agraïment més profund a la meva família. Sense la seva col·laboració hauria estat impossible assolir tots els objectius d'aquest llarg període. Als meus pares Pere i Puri vull agrair el seu suport tant emocional com econòmic i a la Sara li vull agrair de tot cor la santa paciència que ha tingut i com m'ha cuidat durant les meves llargues jornades de treball.

Resum

Avui en dia la manera d'interactuar amb els aparells electrònics està fent que deixem de banda dispositius com el ratolí o el teclat físic per ser completament tàtil. Aquest canvi propicia noves maneres de plantejar el programari.

D'aquí sorgeix Equdia, una aplicació amb una nova manera de representar fórmules matemàtiques orientada a dispositius tàtils. Aquesta representació és jeràrquica i intuïtiva, basada en la primitiva lògica del triangle com a unitat fonamental. L'aplicació està pensada per manipular directament les primitives per a observar la repercussió que tenen aquests canvis en la resta de variables. Està enfocat tant a l'ensenyament de càlcul, com de manera quotidiana com per activitats professionals.

S'han fet proves d'usabilitat i ha tingut bona acceptació, sobretot entre els joves, acostumats a la utilització de mòbils i tabletas, ja que aquest programari requereix un temps d'aprenentatge per a la seva correcta utilització.

Hoy en día la forma de interactuar con los aparatos electrónicos está haciendo que dejemos de lado dispositivos como el ratón o el teclado físico para ser completamente tátil. Este cambio propicia nuevas formas de plantear el software.

De aquí surge Equdia, una aplicación con una nueva manera de representar fórmulas matemáticas orientada a dispositivos táctiles. Esta representación es jerárquica e intuitiva, basada en la primitiva lógica del triángulo como unidad fundamental. La aplicación está pensada para manipular directamente las primitivas para observar la repercusión que tienen estos cambios en el resto de variables. Está enfocado tanto a la enseñanza de cálculo, como de manera cotidiana como para actividades profesionales.

Se han hecho pruebas de usabilidad y ha tenido buena aceptación, sobre todo entre los jóvenes, acostumbrados a la utilización de móviles y tabletas, ya que este software requiere un tiempo de aprendizaje para su correcta utilización.

Nowadays the way we interact with electronic devices is causing side stop devices like mouse or physical keyboard to be completely touchable. This change encourages new ways of develop software.

This raises Equdia, an application with a new way of representing mathematical formulas oriented touch devices. This hierarchical representation is intuitive, based on the primitive logic of triangle as the fundamental unit. The application is designed to directly manipulate primitives to observe the impact of these changes in the other variables. It focuses both teaching calculus, so as to everyday, so as professional activities.

Its usability has been tested and it has been well received, especially in the young groups, accustomed to use phones and tablets, because this software requires a learning time for its use.

Índex

Portada	1
Agraïments	2
Resum	3
Índex	4
Introducció	5
Anàlisis	8
- Representació analítica	8
- Representació quantitativa	10
- Planificació	13
- Presupost	14
- Casos d'ús	15
Disseny	30
- Interfície	30
- Classes	31
- Detalls del projecte	34
- Interfície	36
Resultats	37
- Paràmetres	38
- Exemple pràctic	39
Conclusions	44
Referències	45
Anex 1: TicSalut	46
Anex 2: Festa de la ciència	48

Introducció

Des dels principis de la història, l'ésser humà ha tingut la necessitat de fer càlculs complexos per resoldre innumerables situacions de la vida quotidiana. Es tenen evidències de la utilització d'aparells de càlcul, com l'àbac [Figura 1], centenars d'anys abans de Crist.



Figura 1: Àbac, antic instrument de càlcul.

L'evolució que segueixen els aparells i els elements que utilitzen aquests per representar els càlculs cada cop són més amigables per a les persones. La tendència és apostar per la intuïció i per la encapsulació, de manera que cada cop es treballi amb unitats a més alt nivell. Al film EVA [Figura 3], dirigida per Kike Maíllo al 2011, es pot veure una visió futurista que mostra un futur en el que la programació es fa afegint mòduls encapsulats a l'aplicació per afegir funcionalitats. Aquesta filosofia ja l'ha portat a terme Google amb l'aplicació App Inventor [Figura 2] que permet crear aplicacions per a la plataforma Android sense necessitat de programar ni una sola línia de codi, només cal afegir a l'aplicació que es crea els mòduls amb les accions que es desitgin. D'aquesta manera s'intenta no només que la interacció amb els dispositius sigui més intuïtiva sinó que els objectes amb els que treballem siguin més ben identificables pel que representen. Aquesta evolució també la podem veure en els llenguatges de programació, on cada cop són més propers al llenguatge natural amb llenguatges com Python o Ruby, on s'intenta que el programador pugui escriure el codi de la manera més familiar possible.

En aquest moment la manera que tenim les persones d'interactuar amb els aparells tecnològics que ens envolten està canviant. Estem passant d'utilitzar elements intermediaris com el ratolí o el teclat a modificar allò que veiem a les pantalles de manera tàctil.



Figura 2: Aplicació creada amb App Inventor

Aquest canvi de mentalitat i de relació amb els aparells electrònics com són els *smartphones* o les tablettes ens permet noves formes de plantejar la representació de tot allò que veiem adaptant-ho a aquesta nova manera d'interacció. Amb aquest nou enfocament al cap s'ha portat a terme aquest projecte. Una nova manera de representar i d'interactuar amb les fórmules matemàtiques.

Fins ara la base dels programes de càlcul eren els nombres, aquests formaven la part fonamental, l'entrada de dades als programes eren un conjunt de fórmules i números a partir de les quals es formaven representacions amb tot tipus de gràfiques. Amb aquesta nova visió, en canvi, es dóna una volta a aquest plantejament, fent que l'usuari dissenyi i manipuli gràfiques i obtingui com a resultat la quantificació d'aquestes.



Figura 3: Captura del programador de la pel·lícula EVA.

Equdia és un programa de càlcul basat en el fractal triangular de Sierpinski on, mitjançant les posicions dels triangles a la pantalla i les seves mides, es poden obtenir relacions entre variables i constants. D'aquesta manera es pot obtenir qualsevol tipus de fórmula matemàtica, amb l'avantatge de que la manipulació d'aquestes fórmules és molt intuïtiva i es poden veure els canvis quantitatius a temps real.

L'aplicació, donat que precisa d'espai per poder crear equacions amb certa complexitat, es desenvoluparà per tabletas. Després d'avaluar els diferents sistemes operatius per mòbils més utilitzats, Android, iOS, Blackberry, Windows Phone i Symbian, s'ha decidit per desenvolupar aquest software per la plataforma Android per ser el sistema més estès del mercat, amb una quota de mercat mundial del 25% i superant el 70% a Espanya [Figura 4]. També és important destacar que la Universitat de Barcelona ha fet una forta aposta per aquest sistema operatiu afegint l'ensenyament de programació per a dispositius amb aquesta plataforma i adquirint tabletas Motorola Xoom amb Android 3.2 actualitzable a Android 4.0.1 per a que els seus alumnes puguin desenvolupar-hi. Aquest projecte es portarà a terme usant aquesta tableta.

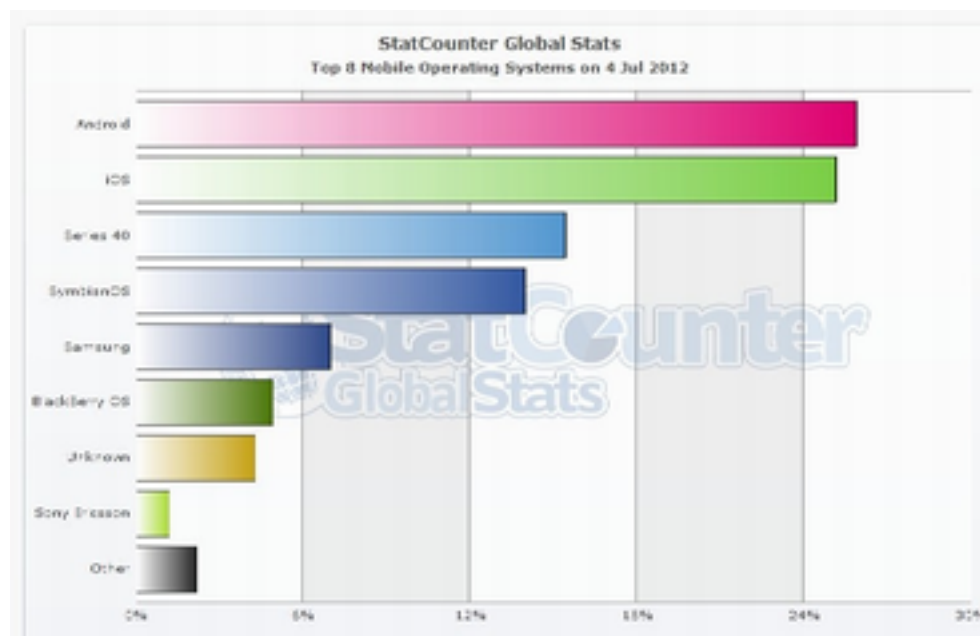


Figura 4: Gràfic de percentatges de dispositius que utilitzen cada SO

A les següents pàgines s'explicarà totes les fases de desenvolupament de l'aplicació. Començant pel plantejament, el modelat, la construcció i acabant amb un petit reportatge dels esdeveniments on ha sigut exposat el projecte a públic tant especialitzat com familiar.

Anàlisi

Amb aquesta aplicació es pretén proporcionar un nou sistema de representació de fórmules matemàtiques interactiu. Al estar orientat a la interacció mitjançant dispositius tàctils, la plataforma escollida ha sigut Android, per ser un sistema operatiu lliure per a mòbils i tabletas principalment, tot i que a l'actualitat ja es comencen a veure televisors i ordinadors personals amb les versions més noves d'aquest sistema operatiu.

A les següents pàgines s'explicarà la evolució que ha seguit el plantejament d'aquesta aplicació des dels seus inicis fins els plans de futur que es tenen per ella. Es parlarà de les bases de Equidia d'una manera més pràctica i es mostraran tots els requeriments que la primera versió d'aquest programa pretén contenir, ja que al ser una aplicació amb grans possibilitats d'expansió, poden haver futures versions amb més opcions.

Representació analítica

L'evolució del sistema de representació del que tracta aquest treball es pot dividir en dos parts diferenciades. En un primer moment es va plantejar una representació purament jeràrquica basada en el fractal de Sierpinski a la que hem anomenat **representació analítica**.

Un exemple d'estructura jeràrquica bàsica seria la divisió que fem dels números. Aquesta divisió agrupa els números en conjunts, i aquests, en nous conjunts. Aquesta representació, típicament feta amb diagrames de Venn, resulta més intuïtiva i fàcil d'analitzar amb casos que requereixen molta profunditat. En la següent imatge [figura 5] es mostra l'exemple de la classificació dels nombres amb un diagrama de Venn.

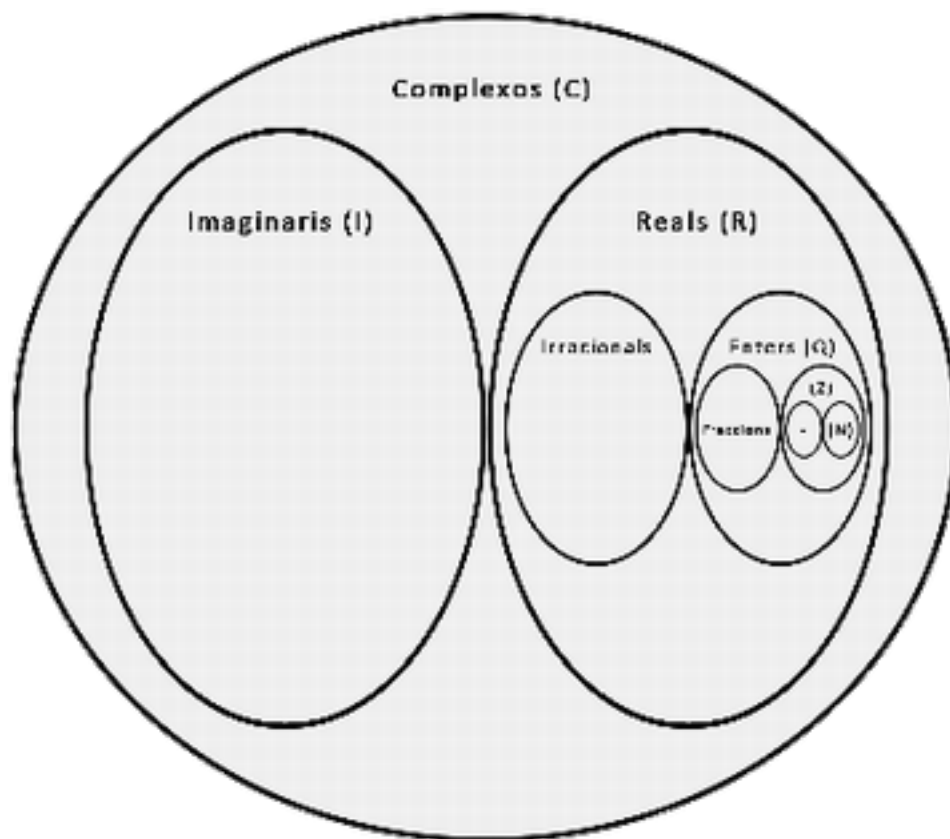


Figura 5: Diagrama de Venn sobre la classificació dels nombres.

Tot seguit es mostra el mateix exemple amb un diagrama triangular [figura 6]. Amb aquesta representació es classifiquen conjunts d'elements en diferents altures, sabent que cada subconjunt representat per un triangle està inclòs dins del triangle superior i, alhora, està format pels triangles per sota d'ell.

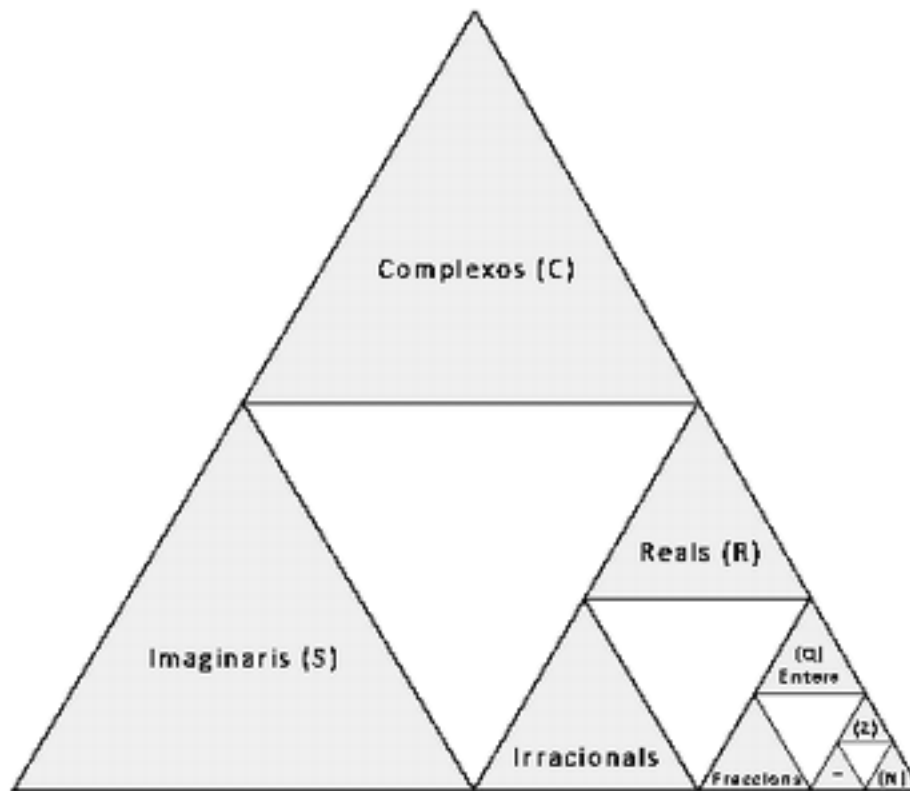


Figura 6: Diagrama d'Equidia sobre la classificació dels nombres

Aquesta versió és estàtica. Els triangles poden contenir valors numèrics que els quantifiquin però, al no estar relacionades ni la forma ni la mida dels triangles amb el valor que aquests representen, és una manera visualment poc clara de representar relacions entre valors. D'aquí va sorgir la idea de la **representació quantitativa**.

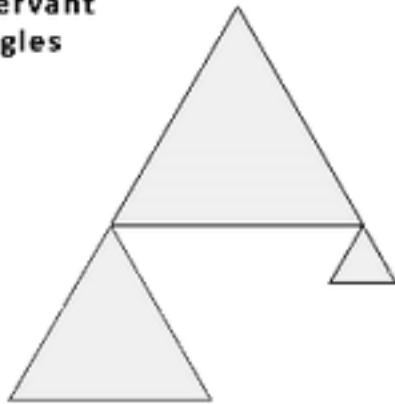
Representació quantitativa

Aquest pas pretén portar més enllà la representació analítica quantificant-la visualment. Això es fa relacionant el valor dels conjunts i elements amb la mida dels triangles. Els triangles passen de tenir una relació merament jeràrquica a tenir una relació de quantitat en la que l'àrea d'un triangle al nivell x , o pare, és la suma de les àrees dels triangles al nivell $x-1$, o fills, procedents d'aquest.

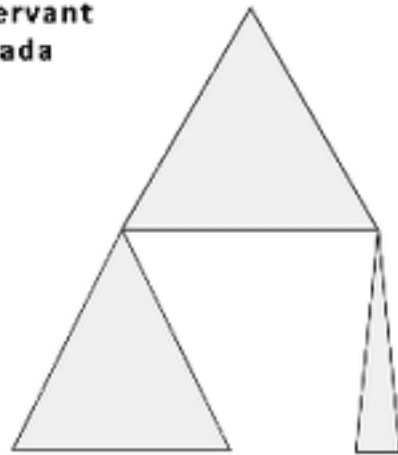
Per aquest propòsit hi ha quatre possibles maneres de representar els triangles respectant que la suma de les àrees dels triangles fills sigui igual a l'àrea del triangle pare [figura 7].

1. Conservant els angles: D'aquesta els triangles fills tindran els mateixos angles i per tant la mateixa forma però variaran en escala.
2. Conservant l'alçada: Els triangles fills conserven l'alçada i varien la seva amplada.
3. Conservant l'amplada: Els triangles fills conserven l'amplada i varien la seva alçada
4. Coincidint vèrtex: L'amplada i l'alçada dels triangles fills és la justa per a que els vèrtex interiors coincideixin.

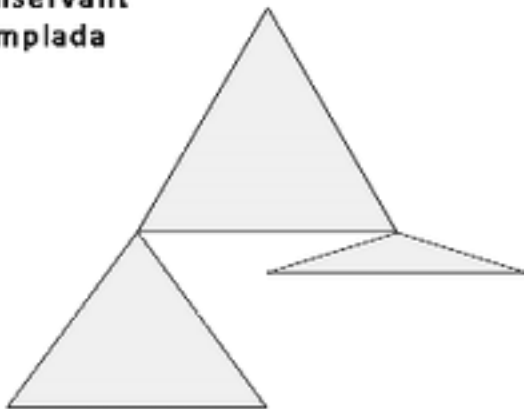
**Conservant
angles**



**Conservant
alçada**



**Conservant
amplada**



**Coincidint
vèrtex**

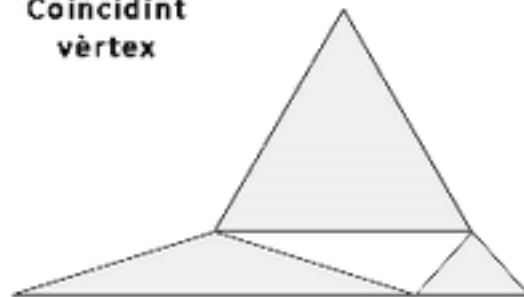


Figura 7: Quatre possibles representacions de la relació d'àrees.

D'aquestes quatre representacions, la més apropiada a l'hora d'esser manipulada modificant les àrees dels triangles és conservant les alçades. D'aquesta manera cada triangle es manté al nivell que li pertoca i la jerarquia es veu molt més clara. De les altres maneres, al variar l'alçada,

els triangles dels nivells inferiors es mourien verticalment provocant confusions i, alhora, al variar l'amplada d'una manera tant abrupte, es provocarien solapaments entre triangles que es trobessin a la mateixa alçada.

Si posem com a constant aquesta alçada, es pot acotar el problema. Sabent que la fórmula del triangle és base per alçada dividit per dos, i definim que l'alçada es mantindrà constant a dos, la fórmula resultant indica que l'àrea del triangle serà igual a la seva base. D'aquesta manera es poden simplificar molts càlculs.

A partir d'aquest moment es parlarà indistintament de triangle, node i variable. Ja que la primitiva visual utilitzada és el triangle, el qual és un node dins l'arbre jeràrquic i representa una variable dins de la fórmula matemàtica.

Donat que aquesta representació és quantitativa, un aspecte indubtablement essencial és mostrar el valor de cadascun dels nodes. Aquest valor s'escriurà al centre de cada triangle i serà un nombre real amb dos decimals de precisió. També s'escriurà sobre aquest nombre l'identificador que cada variable ha de tenir. Aquests identificadors estaran classificats en dos grups, nodes intermedis i nodes fulla, i l'ordre vindrà determinat pel nivell on es trobi el node i per la posició que aquest ocupi. Els nodes fulla estaran identificats per les lletres de la a 'a' la 'l' i en cas de necessitar més, s'utilitzaran nombres darrera de la lletra, a1, b1.. per exemple. L'ordre d'assignació d'aquests identificadors començarà des del nivell inferior al nivell superior, i per cada nivell s'aniran assignant d'esquerra a dreta. A la figura 8 es pot observar el primer node que ha d'aparèixer a la pantalla a partir del qual es formarà tot l'arbre de l'equació.

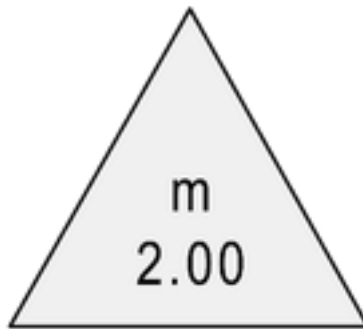


Figura 8: Representació d'un node

Planificació

Un cop definida la teoria base de l'aplicació és el torn de fer la planificació de la feina. Al diagrama de Gantt de la figura 9 es mostren els temps estimats per portar a terme cada apartat durant els sis mesos en els que està planificat el projecte.

Activitat	Gener		Febrer				Març				Abril				Maig				Juny				Juliol	
	11	12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Anàlisi																								
Disseny																								
Implementació																								
Validació																								
Difusió																								
Memòria																								

Figura 9: Diagrama de Gantt amb la planificació del projecte

Durant les primeres setmanes de treball es farà el disseny detallat dels casos d'ús i es començarà a buscar informació sobre les millors eines de treball per desenvolupar l'aplicació, tant el material necessari, com els dispositius, fins a investigar sobre les plataformes possibles per a programar i en el cas d'escollir-ne una, començar a crear l'estructura del projecte i dissenyar classes.

Un cop la base estigui avançada es començarà la implementació. Aquesta estarà dividida en una serie d'iteracions basades en els diferents casos d'ús i per a cada iteració es farà una validació i, el més segur, un nou disseny i/o una nova implementació per solucionar els conflictes que es puguin haver trobat, ja sigui d'errors d'implementació o de disseny que s'han trobat a posteriori.

El dia 1 de Juny se celebra a Girona el TicSalut, que és una fira on es fomenta la comunicació entre empreses que busquen invertir en nous projectes i projectistes que volen rendibilitzar el seu treball. El dia 16 de Juny es celebra la Festa de la Ciència a Barcelona, on es fan una sèrie d'activitats lúdiques per un public familiar. A aquests dos esdeveniments es participarà per fer una divulgació sobre l'Equidia. Aquestes sessions també serviran per continuar fent la validació amb noves persones.

Finalment la tasca es donen tres setmanes com a període per escriure la memòria i deixar el codi polit preparat per l'entrega.

Pressupost

Un cop tenim planificat quant de temps requereix el desenvolupament d'aquesta aplicació es pot definir un pressupost [figura 10] per a que en cas de ser un producte que ens demanés implementar una empresa poguéssim predir quins serien els costos de producció.

Recurs	Cost
Dispositiu (Tablet)	249 €
Equip de treball (PC)	750 €
Programari	0 €
Inscripció TicSalut	90 €
Inscripció Festa de la C.	0 €
Fungibles	100 €
Temps de treball (15€/h)	1.800 €
Total	2.989 €

Figura 10: Pressupost del projecte

El pressupost s'ha fet tenint en compte els següents aspectes:

- La tablet pressupostada és la facilita la universitat al seus alumnes, Motorola Xoom.
- S'ha pensat en un equip de treball de qualitat mitja ja que programar aquest tipus d'aplicacions no requereix molts recursos.
- El programari no té cost ja que s'utilitzaran solucions gratuïtes.
- La participació a la Festa de la Ciència no té preu d'inscripció, tot i així té costos de transport i de temps del personal de divulgació.
- El temps de treball s'ha calculat amb un salari de tècnic de quinze euros per hora durant sis mesos a mitja jornada.

Casos d'ús

Al següent diagrama de casos d'ús [figura 11] es mostren totes les possibilitats que ha d'oferir la primera versió d'aquesta aplicació a l'usuari.

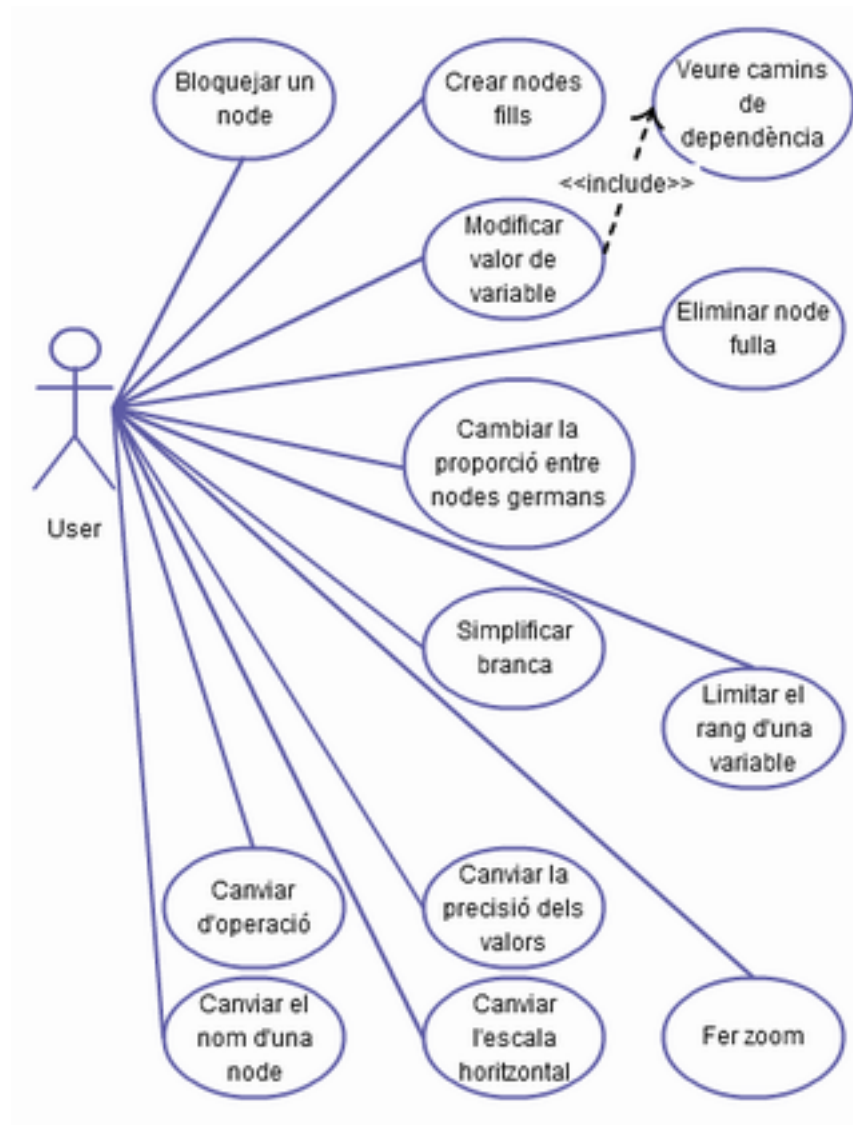


Figura 11: Diagrama de classes de l'aplicació

Un cop arribats a punt, tenint l'estructura base de l'aplicació, s'han de definir els mètodes d'interacció amb les primitives visuals i com es mostraran per pantalla els diferents estats d'aquestes i les seves operacions.

1 - Bloquejar/desbloquejar un node

Cada un dels triangles que es creen representen una variable per defecte, això és així ja que se n'ha de poder canviar el valor des de bon principi. Tot i així ha d'haver la possibilitat de bloquejar una variable fent click sobre el triangle perquè no es pugui canviar el valor mentre estigui bloquejada. Aquests nodes bloquejats es poden interpretar com constants. Les constants vindran definides per una circumferència tangent a les tres arestes del triangle [figura 12].

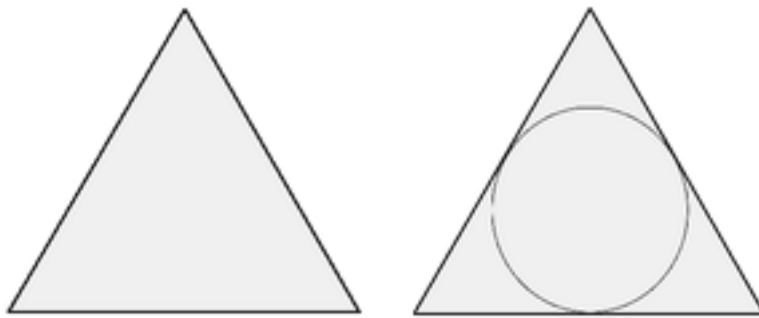


Figura 12: Node bloquejat / Node desbloquejat

Cas d'ús: Bloquejar/desbloquejar una variable
Actor: Usuari
Precondicions: Tenir el programa carregat amb algun triangle visible
Escenari principal: 1. L'usuari fa click sobre la part interna d'un triangle. 2. El sistema dibuixa una circumferència tangent als tres costats del triangle indicant que no es podrà modificar el valor d'aquell node.
Postcondicions: El triangle conté una circumferència tangent i queda bloquejat.

2 - Crear nodes fills

Tota variable ha de poder ser dividida en dos variables filles. Com la suma d'àrees d'aquestes dos variables ha de ser el valor del triangle originari, aquestes variables tindran valors entre zero i el valor del node pare. L'acció per fer-los aparèixer serà arrossegant el dit des de la base del triangle cap avall.

El triangle pare ha de contenir $n-1$ línies divisòries a la seva base, on n és el nombre de fills, indicant les divisions que s'han hagut de fer a la base per formar cadascun dels triangles. Els triangles, per defecte, apareixeran com a bloquejats com a mitjà de control d'errors [Figura 13].

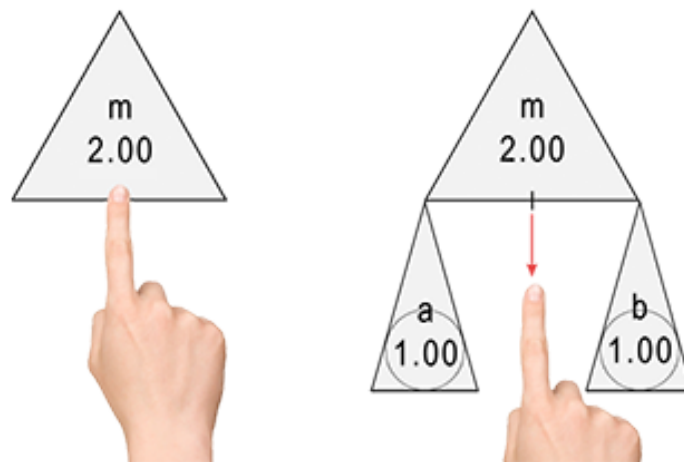


Figura 13: Creació de nodes fills

Cas d'ús: Crear nodes fills
Actor: Usuari
Precondicions: La variable que es vil dividir no té fills i està desbloquejada
Escenari principal: 1. L'usuari toca la base del triangle i arrossega el dit cap avall 2. Apareixen dos triangles que sumen el valor del pare i es troben bloquejats.
Postcondicions: S'han creat dous nos triangles fills del seleccionat i s'ha actualitzat la fórmula amb les noves variables.

3 - Modificar el valor d'una variable

L'usuari podrà canviar el valor de qualsevol triangle tocant el vèrtex superior i arrossegant el dit verticalment per assignar a cada variable el valor desitjat, sempre i quant no es trobi bloquejat [figura 14]. Canviar el valor d'una variable significa que per mantenir la igualtat de l'equació, altres nodes han de canviar el seu valor. Per això, s'ha de garantir que l'usuari no podrà modificar una variable si no existeix un camí de propagació. Aquest camí s'ha de preparar bloquejant i desbloquejant els triangles per tal d'aconseguir un camí correcte i únic.

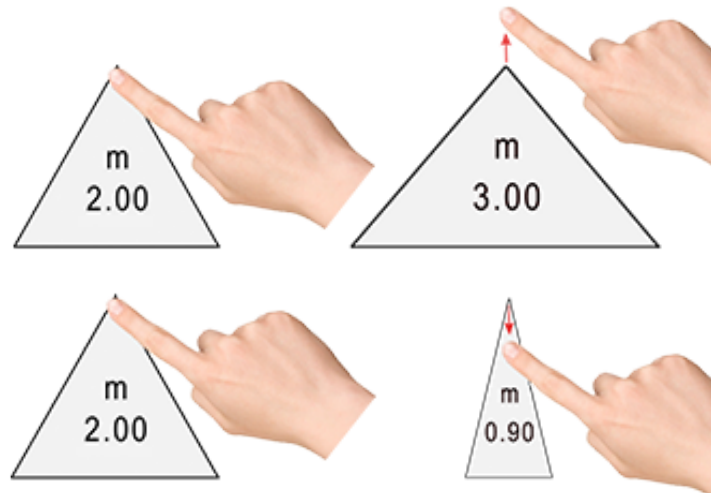


Figura 14: Modificar el valor d'una variable.

Existeixen tres tipus de camins de propagació a tenir en compte:

- a) Cap avall, propagant el canvi cap a un node fill fins arribar a un node fulla.
- b) Cap a dalt, propagant el canvi fins el node arrel.
- c) Cap a un germà. Sempre que no hi hagi camí vertical de propagació, es podrà propagar cap a un únic germà i des d'ell cap a un node fulla.

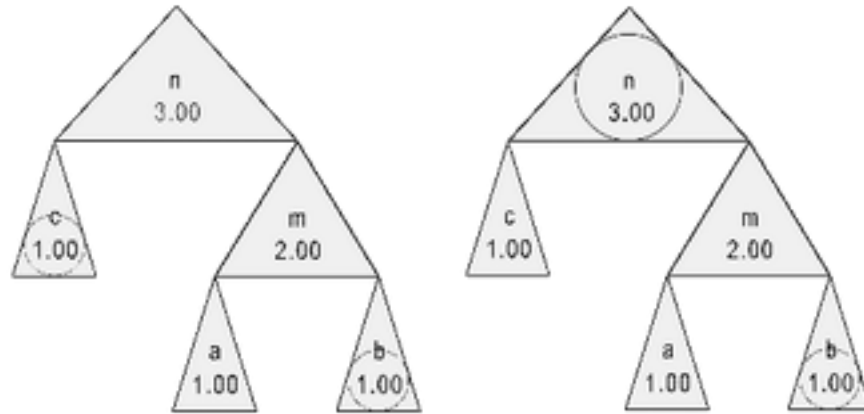


Figura 15: Estats possibles que permeten la modificació d'un node (m)

Sempre s'han de donar la situació 'a' acompanyada de la situació 'b' o 'c'. O sigui, sempre han d'existir un camí inferior i un camí superior, l'inferior cap a un dels fills, si n'hi ha, i el superior pot ser cap al node pare o cap a un node germà.

Al intentar modificar el valor d'una variable s'ha de mostrar el camí de propagació existent o, si no hi ha un únic camí, es mostrarà el camí fins al punt on hi ha el conflicte. És important destacar que l'inici del camí es marcarà un quadrat i el final amb una estrella de deu puntes.

Al canviar el valor d'una variable es pot voler canviar molt depresa per arribar a valors molt grans o, pel contrari, variar de manera milesimal aquests valors per indicar quantitats molt precises. A l'etapa de disseny es trobarà una solució pràctica per aquest problema. També és necessari comentar que per denotar el poc valor de les variables i constants que tinguin el seu valor al rang $(-1, 1)$ el gruix de les seves arestes haurà d'ésser menor, com es mostra a la [figura 15].

Cas d'ús: Modificar el valor d'una variable
Actor: Usuari
Precondicions: El node no està bloquejat i existeix un camí de propagació únic.
Escenari principal: 1. L'usuari toca el vèrtex superior del triangle i arrossega el dit verticalment. 2. El triangle es va fent més ample o més estret i el valor d'aquest va variant en conseqüència.
Postcondicions: La variable pren el valor i la mida que pertocquen i tots els triangles al camí pateixen els canvis oportuns per conservar la coherència de l'arbre.

4 - Eliminar un node fulla

S'ha posat la restricció de que només els nodes fulla es puguin eliminar. S'ha fet així per evitar eliminar tota una branca per error. Per eliminar un node fulla, així com al modificar el valor d'una variable, ha d'haver un camí de propagació per poder mantenir correcta la fórmula resultant. L'acció per portar a terme l'eliminació és arrossegar el dit des de la base del triangle cap amunt [figura 16].

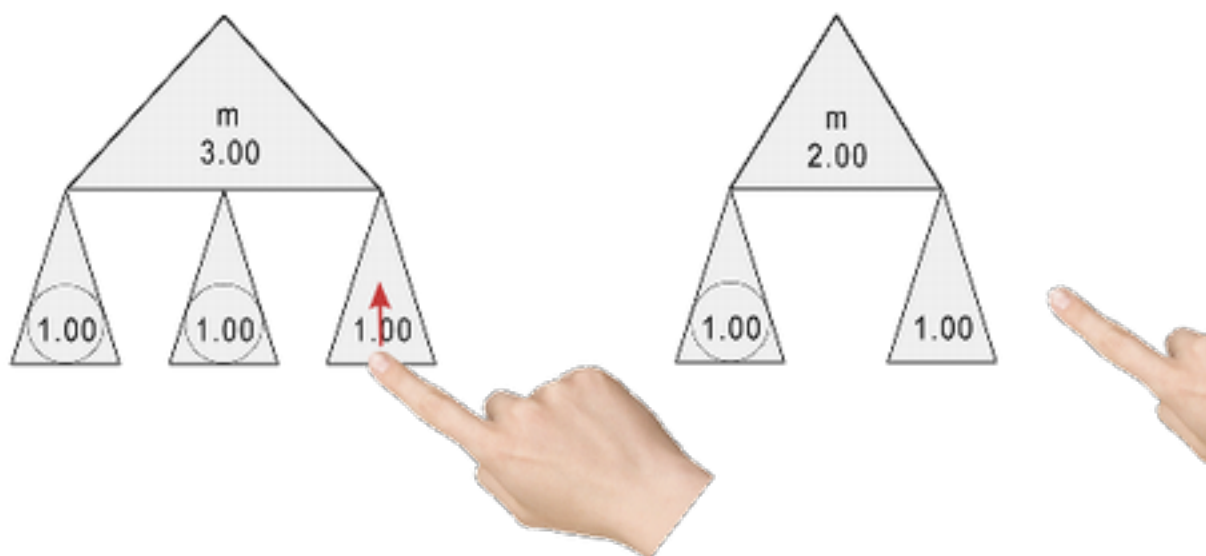


Figura 16: Eliminació d'un node fulla.

Cas d'ús: Eliminar un node fulla
Actor: Usuari
Precondicions: El node no està bloquejat i existeix un camí de propagació únic.
Escenari principal: 1. L'usuari toca la base del triangle i arrossega el dit cap amunt. 2. L'arbre s'actualitza sense el node eliminat
Postcondicions: El triangle queda eliminat i tots els triangles al camí han patit els canvis oportuns per conservar la coherencia de l'arbre.

5 - Canviar la proporció entre nodes germans

Donat que al crear dos triangles fills d'un altres triangle no és possible determinar amb exactitud el valor que aquests han de prendre en aparèixer, una manera per poder variar el seu valor, a part de la manera que ja coneixem, s'ha de poder moure la línia divisòria que hem comentat a l'apartat "Crear nodes fills" per tal de canviar la proporció d'àrea que cadascun dels fills pren del node pare [figura 17].

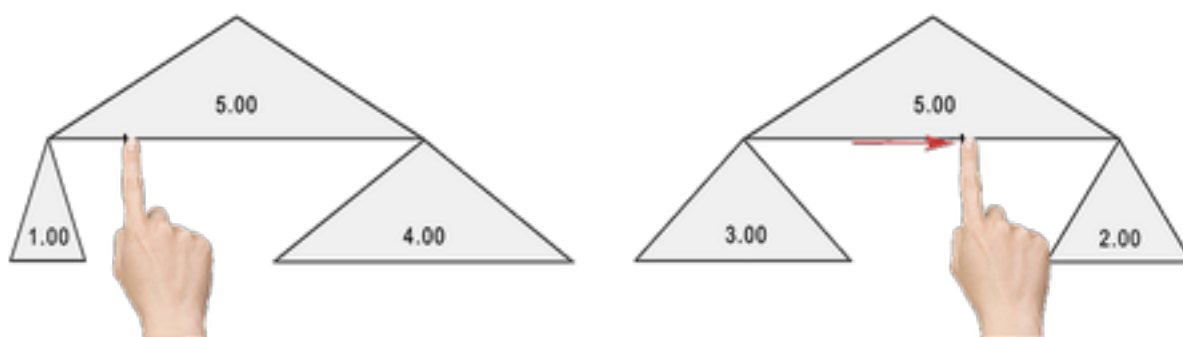


Figura 17: Cambi de proporció entre ndes germans

Cas d'ús: Canviar la proporció entre nodes germans
Actor: Usuari
Precondicions: Els dos triangles estan desbloquejats i existeix un camí de propagació per a cada un d'ells.
Escenari principal: 1. L'usuari toca la línia divisòria a la base del triangle i arrossega el dit horitzontalment. 2. A mida que es desplaça el dit s'actualitza la mida del nodes germans i dels seu descendents en temps real.
Postcondicions: Els triangles han el valor i la mida segons la proporció indicada i tots els triangles als camí han patit els canvis oportuns per conservar la coherència de l'arbre.

6 - Simplificar una branca

En qualsevol moment pot ser que sigui necessari que una variable estigui formada per múltiples variables, per aquest motiu un arbre binari com el plantejat fins al moment és una limitació massa gran. Ha d'haver la possibilitat de fer que tots els nodes fulla a partir d'un cert triangle es posin directament com a fills d'aquest. En fer això la fórmula no es veu modificada, només s'eliminen els nodes intermedis que poden mancar de sentit o d'importància. Per aconseguir això s'ha de fer el mateix gest que per eliminar un node fulla, tocar la base del triangle del qual volem simplificar la descendència i arrossegar el dit cap amunt [Figura 18].

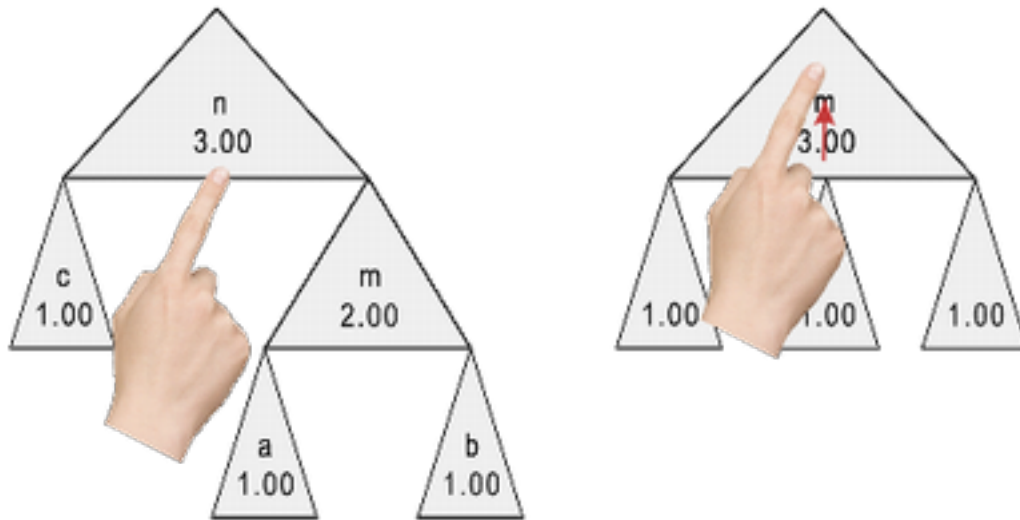


Figura 18: Simplificació d'una branca.

Cas d'ús: Simplificar una branca.
Actor: Usuari
Precondicions: El triangle es troba desbloquejat i no és un node fulla.
Escenari principal: 1. L'usuari toca la base del triangle i arrossega el dit cap amunt. 2. El sistema col·loca els nodes fulla derivats del node seleccionat com a fills directes d'aquest.
Postcondicions: Tots els nodes fulla derivats del triangle seleccionat ara pegen d'aquest.

7 - Limitar el rang d'una variable

S'ha de poder restringir el rang de valors que podrà prendre d'una variable. Això es mostrarà pintant el perímetre dels triangles de color vermell pels nombres negatius i de color blau per als nombres positius. Ambdós contindran el nombre zero com a possible valor. Per assignar el rang desitjat a una variable s'ha de prémer el vèrtex superior del triangle durant dos segons. Automàticament el triangle pintarà el seu perímetre depenent del signe del valor que tingui en aquell moment. Per treure aquesta limitació i fer que pugui prendre qualsevol valor real només cal tornar a prémer durant dos segons els vèrtex superior.

Per mantenir la regla principal de que cada triangle té la suma de les àrees dels triangles fills, en cas de que la suma dels valors d'aquests sigui menor a la suma de la seva àrea, es dibuixaran dos triangles concèntrics. Un dels triangles, amb línia continua, tindrà com a àrea la suma de les àrees dels fills. L'altre triangle tindrà com a àrea la suma de les àrees dels fills i el seu perímetre es dibuixarà amb línia discontinua. Aquest triangle serà el que representarà la suma en valor absolut dels triangles fills. A l'exemple de la figura 19 podem observar com s'han representat els dos triangles comentats, el resultat de la operació, amb valor 1, i el resultat de fer la operació en valor absolut dibuixat amb el contorn discontinu.

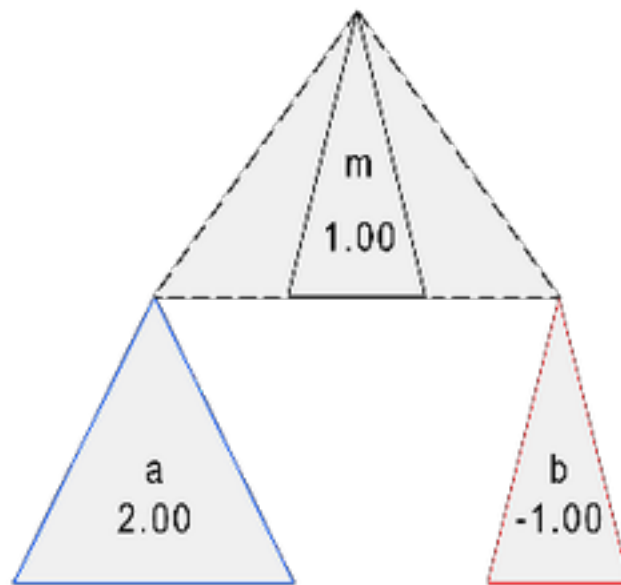


Figura 19: Exemple de triangle representant valor absolut

Cas d'ús: Limitar el rang de valors que pot prendre un node
Actor: Usuari
Precondicions: El triangle es troba desbloquejat.
Escenari principal: 1. L'usuari pressiona el vèrtex superior del triangle durant dos segons. 2. El sistema pinta del color que pertoca el perímetre del triangle seleccionat.
Postcondicions: Els costats del triangle s'han pintat de blau o vermell, depenent del valor del triangle en aquell moment.

8 - Canviar el nom d'una variable

Ja s'ha comentat com s'assignen els noms de les variables per defecte, no obstant, en molts moments l'usuari pot voler canviar el nom d'un node per tenir una referència més clara del que fa referència, això es farà fent click sobre el nom de la variable a la barra lateral de l'aplicació.

Cas d'ús: Canviar el nom d'una variable
Actor: Usuari
Precondicions: Tenir oberta l'aplicació.
Escenari principal: 1. L'usuari pressiona sobre una variable de la llista. 2. El sistema mostra una finestra on l'usuari pot introduir el nou nom. 3. L'usuari introdueix el nou nom i clicka acceptar. 4. El sistema modifica tant a la llista com sobre el triangle el nom de la variable.
Postcondicions: El nom de la variable a la llista i a l'arbre s'ha modificat.

9 - Canviar d'operació

Com a tota eina de càlcul, s'han de permetre nombroses operacions per a tenir un gran ventall de possibilitats. Fins ara s'havia plantejat el mètode només amb la suma i la resta però la primera versió d'aquest software permetrà, a més, la multiplicació i la divisió. Per tant s'ha de poder alternar entre SumRes i MulDiv. Per fer això s'ha de pressionar sobre el centre del triangle dos segons. Cada cop que es faci aquesta acció s'efectuarà un canvi d'operació conformant una rotació entre les operacions disponibles.

Totes les regles explicades anteriorment s'han de mantenir per mantenir la coherència del mètode. Les multiplicació de les àrees filles resultarà l'àrea pare, i aquest, per denotar que està representant la multiplicació, es dibuixarà sobre ell una línia horitzontal i una fletxa vertical com el mostrat a la següent imatge [figura 20].

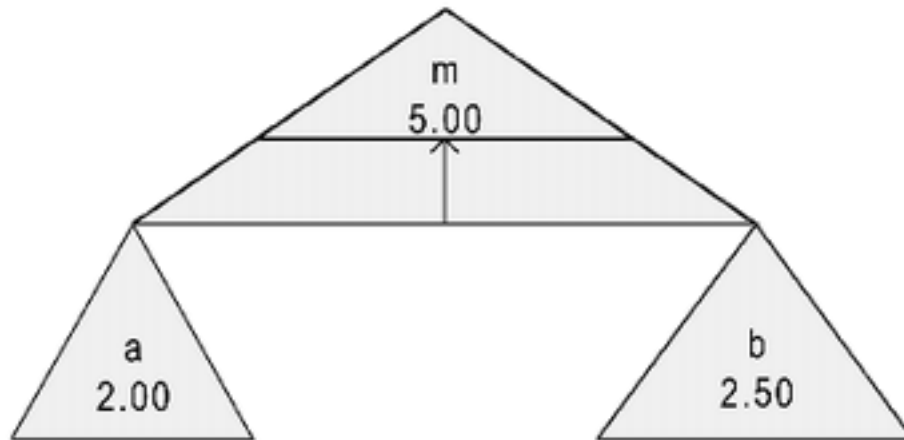


Figura 20: Exemple de node com a resultat d'una multiplicació

Cas d'ús: Canviar d'operació
Actor: Usuari
Precondicions: El triangle no es troba bloquejat, té nodes fills i un camí superior de propagació correcte.
Escenari principal: 1. L'usuari pressiona el centre del triangle durant dos segons. 2. El sistema dibuixa sobre el triangle el símbol que identifica la nova operació i canvia, segons aquesta, els valors del triangle seleccionat i dels seus antecessors.
Postcondicions: El valor i la mida del triangle canvien aplicant la nova operació sobre els valors dels fills d'aquest i es mostra una figura identificativa al centre. A més, canvien de mida i valor tots els nodes afectats per aquest canvi en la variable.

10 - Canviar la precisió dels valors

Es pot voler treballar a diferents escales, tant amb molts decimals i una alta precisió o amb valors sencers tant grans com sigui necessari, per això ha d'haver un mètode per deixar a l'usuari canviar la precisió amb la que es mostren els valors. Per variar el nombre de decimals a mostrar s'ha de tocar el centre del triangle i arrossegar el dit horitzontalment fins trobar la precisió desitjada.

Variar la precisió no només serveix per poder observar més amb més decimals el valor dels nodes sinó que quan es modifiquin els valors d'aquests, el canvi es farà en base a l'últim decimal, per tant, un variable amb tres decimals observables la podrem fer créixer i decreixer mil vegades més poc a poc que una en la que només es mostri la part sencera [figura 21].

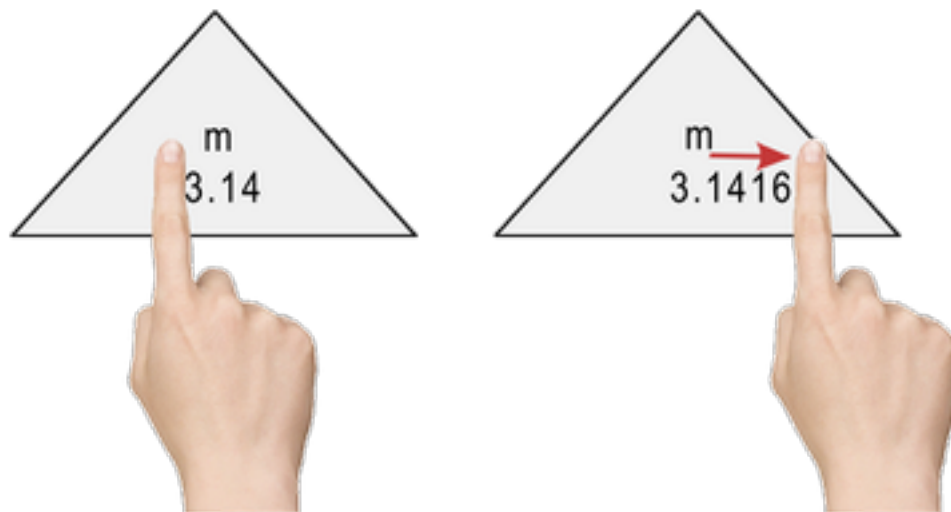


Figura 21: Canvi en la precisió dels valors d'un node.

Cas d'ús: Canviar la precisió dels valors
Actor: Usuari
Precondicions: Tenir el programa carregat amb algun triangle visible
Escenari principal: 1. L'usuari presiona el centre del triangle i desplaça el dit horitzontalment. 2. A mida que es desplaça el dit el sistema va mostrant més o menys decimals.
Postcondicions: El valor del triangle es mostra amb el número de decimals desitjat.

11 - Fer zoom

És possible que donada la limitada superfície dels dispositius tàctils actuals, ràpidament l'arbre que s'estigui representant sobresurti de la part visible de la pantalla. Per aquest motiu s'ha de poder fer *zoom in* i *zoom out*. Així es podrà tant visualitzar l'arbre complet com centrar-se en una certa zona. Per a aquest efecte es col·locaran dos botons a la pantalla.

Cas d'ús: Fer zoom
Actor: Usuari
Precondicions: Tenir oberta l'aplicació.
Escenari principal: 1. L'usuari pressiona el botó de zoom-in o zoom-out. 2. El sistema dibuixa els triangles més grans o més petits i varia també la mida de la font per als valors dels nodes.
Postcondicions: Es visualitzen els triangles i els seus valors augmentats o disminuïts segons el zoom desitjat.

12 - Canviar l'escala horitzontal

Donat que l'altitud dels triangles és fixa, el valor que aquests puguin prendre afecta únicament a la seva amplitud. Per aquest motiu és possible que en treballar amb valors molt grans o molt petits, l'amplitud dels triangles impossibiliti la correcta utilització del software. Per solucionar aquesta complicació s'ha de permetre a l'usuari modificar l'escala horitzontal de la representació de manera que a valors extrems la visualització sigui còmode i pràctica. Per tant, tocant el vèrtex superior de qualsevol triangle i desplaçant el dit horitzontalment veurem com va variant l'amplitud amb la que es mostren totes les figures.

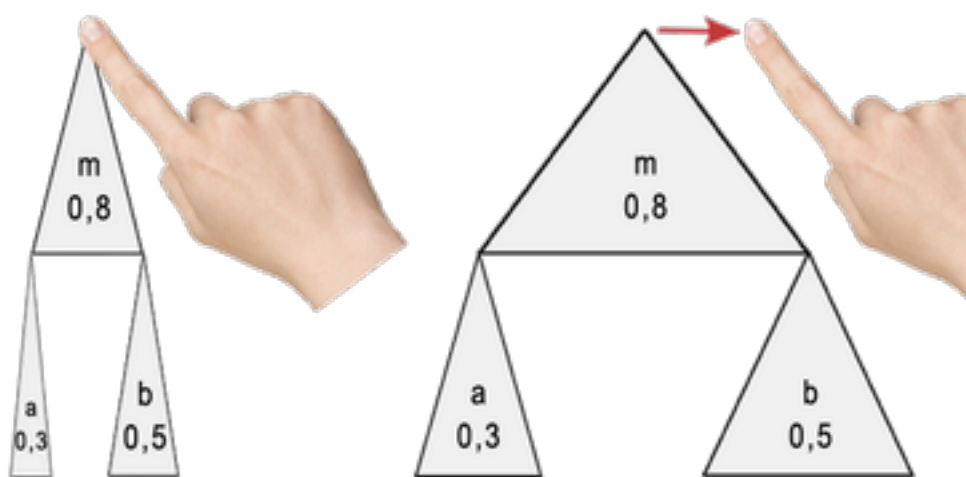


Figura 22: Canvi en l'escala horitzontal

Cas d'ús: Bloquejar una variable
Actor: Usuari
Precondicions: Tenir oberta l'aplicació.
Escenari principal: 1. L'usuari pressiona el vèrtex superior d'un triangle i es desplaça horitzontalment. 2. El sistema fa més amples o estrets tots els triangles a mida que es desplaça el dit.
Postcondicions: Els triangles es visualitzen amb l'amplitud proporcional segons assignada.

Disseny

Un cop explicada la part teòrica del mètode de representació utilitzat i havent especificat els requeriments és hora de determinar la implementació de l'aplicació. En aquest apartat es mostraran els diagrames utilitzats per programar així com exemples d'algorismes implementats.

Interfície

El primer que anem a especificar serà l'aspecte visual de l'aplicació. Donat que es poden crear arbres molt grans per a equacions llargues o complexes, una gran part de la pantalla ha d'estar destinada a aquest recurs. També és important mostrar en una posició preferent la equació que s'ha dissenyat a partir de l'arbre. A més, per a poder consultar el valor de tots els nodes d'una manera més ràpida, es reservarà una àrea lateral on es mostri una llista amb el nom del node seguit del seu valor. Per tant, fent un mockup que compleixi aquests requisits aconseguim que el disseny de la interfície quedi de la següent manera [figura 23]:

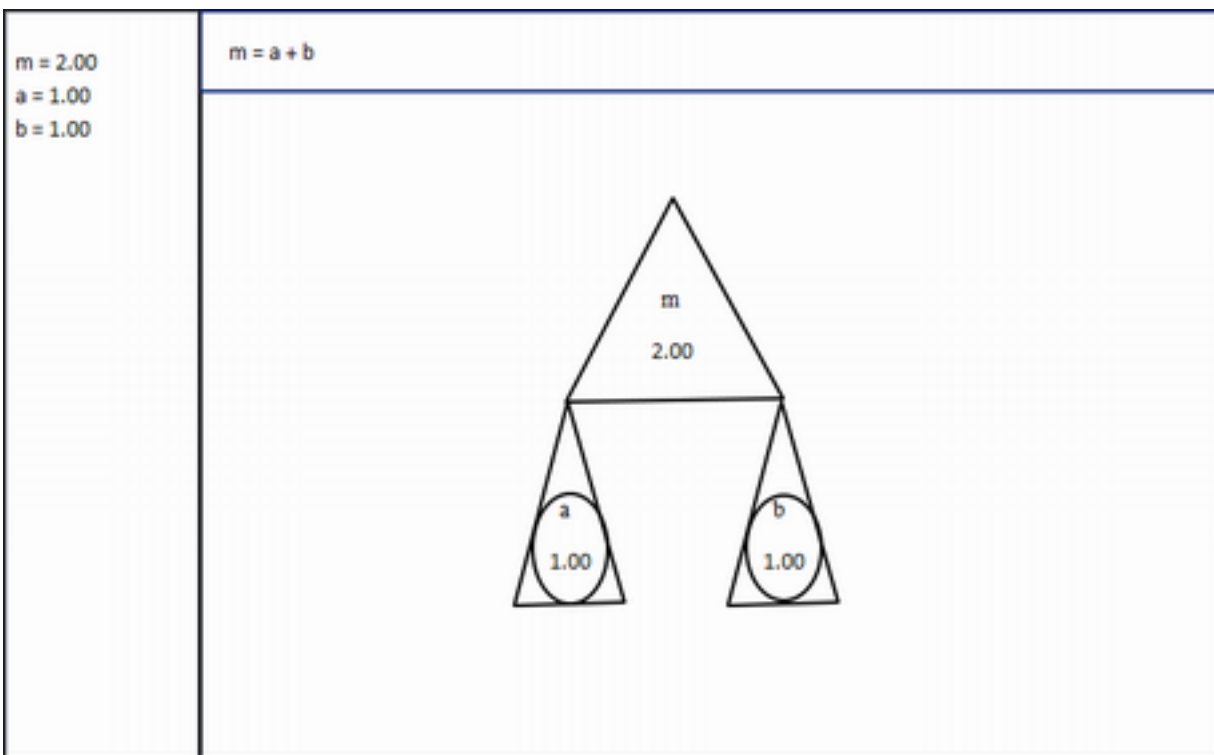


Figura 23: Mockup de la finestra principal de l'aplicació

Només definim una pantalla ja que s'intenta no haver d'interactuar amb menús on s'hagin d'entrar números manualment, seleccionar checkboxes ni cap altra acció a la que estem acostumats però és poc representativa de l'acció que volem realitzar.

Classes

En aquest apartat es definiran les classes i els mètodes a utilitzar tenint en compte la plataforma per a la que es desenvoluparà l'aplicació. La plataforma és Android, sistema operatiu propietat de Google. Un gran avantatge de programar per aquest sistema operatiu radica en que el software de desenvolupament és completament gratuït. Tot i la multitud de possibilitats que se'ns ofereixen a l'hora de voler crear una aplicació per Android, com són:

1. DroidDraw: Programa lliure pel disseny d'interfícies per Android.
2. App Inventor: Software gratuït que permet la programació de codi mitjançant la col·locació de mòduls que identifiquen accions que podrà realitzar l'aplicació.
3. Mono: Plataforma Open Source per programar aplicacions multiplataforma (iOS i Android) en C# amb un framework propi.
4. Basic4Android: Plataforma per desenvolupadors especialitzats en el llenguatge Visual Basic. Té un preu de 49 euros la versió *light* i 249 la versió completa. Les diferències entre versions són llibreries com per exemple per poder accedir a recursos com NFC.
5. LiveCode: Programa per desenvolupar aplicacions multiplataforma (Android, iOS, Windows Phone, Linux i Web). Utilitza un llenguatge de programació propi i, pel seu us, es necessita una llicència que té un cost de 99 dòlars.
6. ADT Plugin for Eclipse: Plugin oficial desenvolupat per Google per Eclipse i completament gratuït. El llenguatge utilitzat és Java.

Després de comparar les diferents possibilitats la plataforma escollida per desenvolupar l'aplicació serà Eclipse amb el plugin ADT (Android Development Tools), ja que tot i no ser multiplataforma, és gratuït, utilitza un llenguatge àmpliament conegut i té les últimes SDKs gracies a que és la mateixa empresa que actualitza el sistema operatiu la que dóna suport.

Un cop havent decidit la plataforma sobre la que es treballarà i havent fet un treball de recerca sobre quines són les apis que millor s'adapten a les nostres necessitats s'han de determinar les classes que ens faran falta per complir el nostre objectiu. Tot seguit es mostrarà el diagrama de classes i una explicació de les apis i dels objectes utilitzats.

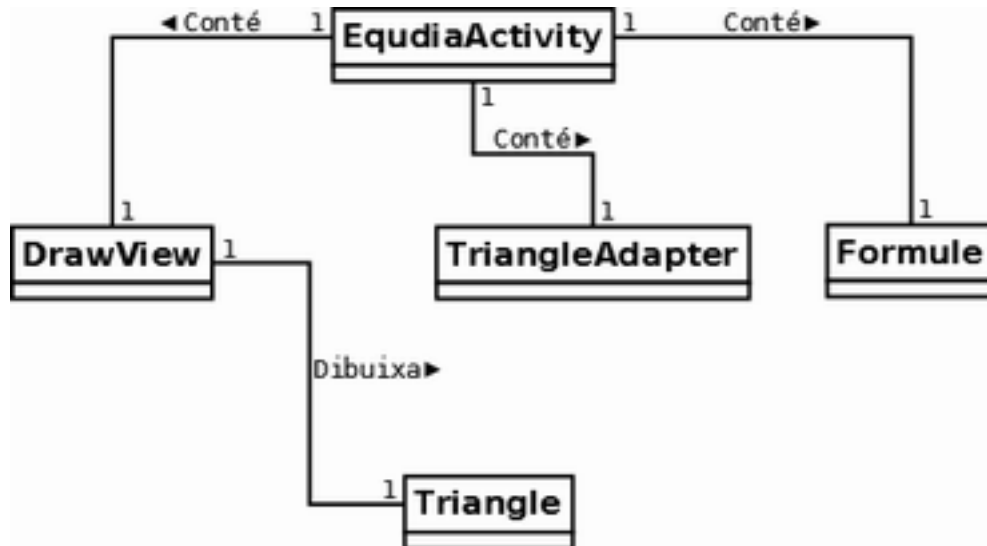


Figura 24: Diagrama de classes

Com es pot observar, tenim com a centre neuràlgic **EquidiaActivity**, que és la classe que defineix la interfície principal. D'aquesta en deriven tres classes que corresponen a cada un dels mòduls en el que es divideix la finestra. Finalment la classe **Triangle** és la que instancien cada un dels nodes de l'arbre.

Descripció de cada una de les classes:

- EquidiaActivity:** És la classe principal i hereta d'**Activity**. Una activity és una classe que carrega una serie de widgets o components a la pantalla, ja sigui dinàmicament o per xml (més eficient però menys versàtil). Aquesta classe farà de controlador i d'intermediari entre les tres vistes. Serà l'encarregat de rebre l'arbre de nodes quan es produeixi una modificació que hagi d'afectar a una vista i li enviarà aquest arbre.
- DrawView:** Aquesta classe estén de **View** i s'ha personalitzat completament i utilitza mètodes de la classe canvas de java per customitzar la manera de mostrar les components per pantalla i el refresc d'aquesta. **DrawView**

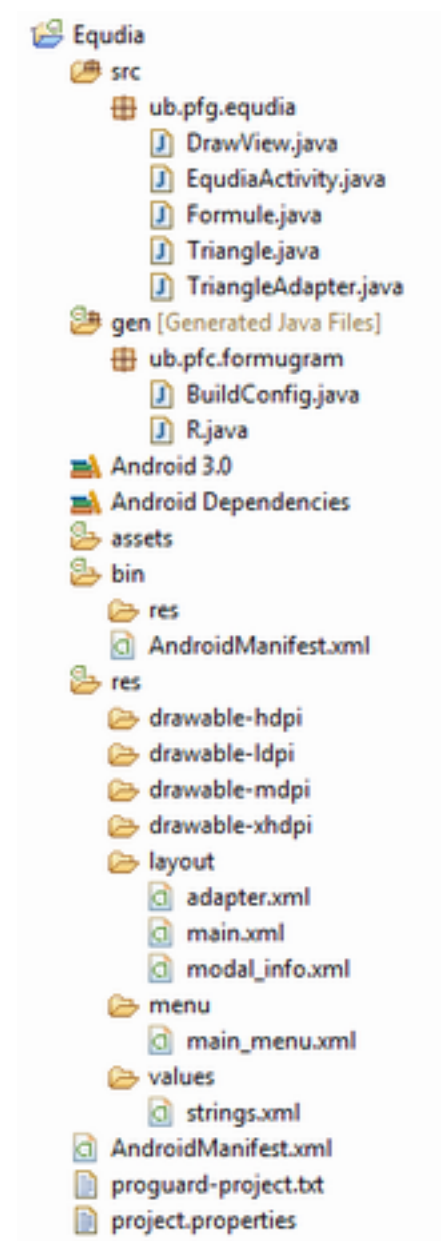
implementa tres interfícies dedicades a la gestió d'events tàctils.

Aquestes interfícies són:

- i. **OnClickListener**: Gestor de l'event onClick. Sempre que es faci un click s'ha de buscar quin és el punt clicat i si es troba a l'interior d'algun dels triangles. L'únic cas d'ús que utilitza aquest event és bloquejar un node.
- ii. **OnLongClickListener**: Gestor de l'event onLongClick. Aquest event s'utilitza en els casos d'ús limitar el rang d'una variable i canviar d'operació.
- iii. **OnTouchListener**: Aquesta interfície proporciona nombroses funcions pel control d'events. Amb l'event de tipus MotionEvent es pot gestionar quan l'usuari ha tocat la pantalla, quan ha arrossegat el dit i les coordenades de la nova posició i quan ha separat el dit de la pantalla. Aquest mètode, que ocupa més de tres-centes línies, controla en tot moment l'estat en el que està l'usuari, això és, on ha clicat, cap a on s'està desplaçant i a quines funcions cridar en cada cas.
En aquesta funció és on s'utilitzen més els llistats i les constants per distingir quan l'usuari, per exemple, s'ha desplaçat o a fet LongClick, ja que l'event ACTION_MOVE salta només que hi hagi un moviment d'un píxel, i per tant aquesta situació s'ha de controlar.
- c. **Triangle**: És la classe que implementen tots els nodes de l'arbre. Aquesta classe conté totes les variables que es necessiten de cada node com el valor, la posició dels vèrtex, el rang de valors possibles, etc. També conté més de cinquanta mètodes per tractar les posicions, les mides, les escales, els valors i les relacions entre els nodes, ja que es guarda una relació pare-fills i una relació fill-pare. Això és perquè s'ha de recórrer l'arbre cap adalt i cap abaix i, donat que el programa no consumeix pràcticament recursos de memòria, s'intenta optimitzar la velocitat a les operacions per tal de garantir un bon refresc de pantalla.
- d. **TriangleAdapter**: Aquesta classe serveix per carregar a cada posició de la llista lateral cada un dels nodes de la forma *nom = valor...*
- e. **Formule**: Aquesta classe és l'encarregada de calcular la fórmula resultant de l'arbre de nodes.

Detalls del projecte

Un cop creat el projecte, passem a explicar de quines carpetes, packages, classes i demés arxius es compona l'aplicació desenvolupada amb el plugin d'Android Development Tools sobre Eclipse. A la figura [XX] es pot observar l'arbre de directoris en el que es compona el projecte.



src: En aquesta carpeta és on van els packages amb totes les classes de l'aplicació. en aquest cas, donat que la complexitat ve donada per l'algorísmica però no hi ha un gran nombre de classes, només hi ha un package.

gen: En aquesta carpeta es contenen les classes que utilitza l'ADT per a compilar l'aplicació. Aquests fitxers es creen automàticament.

Android 3.0 i Android Dependencies: Aquí és on es contenen les llibreries de tipus jar que utilitza l'aplicació. En aquest cas no es necessita cap llibreria externa, tot és codi propi.

assets: En aquesta carpeta és on es guarden recursos que no siguin imatges, ja que aquestes tenen el seu directori propi. Es poden guardar, per exemple, fonts alternatives o arxius d'àudio.

bin: Aquest directori conté l'AndroidManifest. Es tracta d'un arxiu XML que conté la configuració del projecte. En aquest arxiu s'especifica la versió d'android mínima per poder executar l'aplicació, el nom de l'aplicació, per quan aquesta sigui instal·lada, l'activitat principal que s'executarà en obrir el programa i d'altres configuracions com els permisos d'accés a internet, a NFC i d'altres.

res: Aquest directori està destinat a l'apartat visual. En les carpetes que comencen per drawable és on es guarden les imatges. La mida de les imatges es mesura en dpi

(Dots

Per Inch) i cada una de les carpetes, segons la inicial, ha de

Figura 25: Directoris del projecte

contenir imatges en diferents resolucions, High, Large, Middle i Extra High. Si es col·loquen les imatges amb el mateix nom en diferents resolucions, l'aplicació per si sola carregarà les imatges de la mida correcta segons la resolució i la mida de la pantalla del dispositiu on s'instal·li.

A més de contenir les imatges, en aquest directori existeixen tres carpetes que contenen XMLs. Aquestes carpetes són:

- **Layout:** Aquí es guarden els XML que defineixen els widgets de la interfície i totes les seves propietats. En aquesta aplicació tenim dos XMLs en aquest directori, un d'ells és main.xml, amb el qual es configura la pantalla principal amb les tres divisions que hem definit al començament del disseny. L'altre XML és adapter.xml. Sempre que es vulguin editar els elements d'una llista s'ha de crear un adapter, aquest conté la configuració dels elements d'una posició de la llista. En carregar la llista, es carregarà cada posició segons la configuració d'aquest adapter.

- **Menu:** En aquesta carpeta hi ha un sol XML. Aquest arxiu defineix com serà el menú superior que Android dona possibilitat que aparegui a les aplicacions.

- **Values:** Aquest directori també conté només un sol arxiu, strings.xml. En aquest arxiu, com el seu nom indica, es defineixen totes les cadenes que es necessiten, des del nom de l'aplicació fins a totes les paraules que han d'aparèixer. Opcionalment es poden crear les carpetes values-xx on es defineixen les paraules en diferents idiomes on xx és l'idioma utilitzat, (ca: català, en: anglès, it: italià, etc). En la nostra aplicació no hi haurà paraules ja que s'intentarà prescindir d'aquestes i donar missatges d'una manera més visual i interactiva.

Finalment ens trobem amb tres arxius fora dins del directori principal del projecte. Això és perquè són arxius de configuració. Primerament tenim AndroidManifest.xml, que és una l'original a partir del qual l'Eclipse crea automàticament la copia que hem trobat anteriorment dins de la carpeta bin. Tot seguit ens trobem proguard-project.txt. Aquest fitxer serveix per indicar si necessitem que el codi s'ofusqui automàticament al compilar l'aplicació. I finalment trobem project.properties, on es defineix quina és la versió d'SDK d'Android amb la que es vol compilar l'aplicació.

Interfície

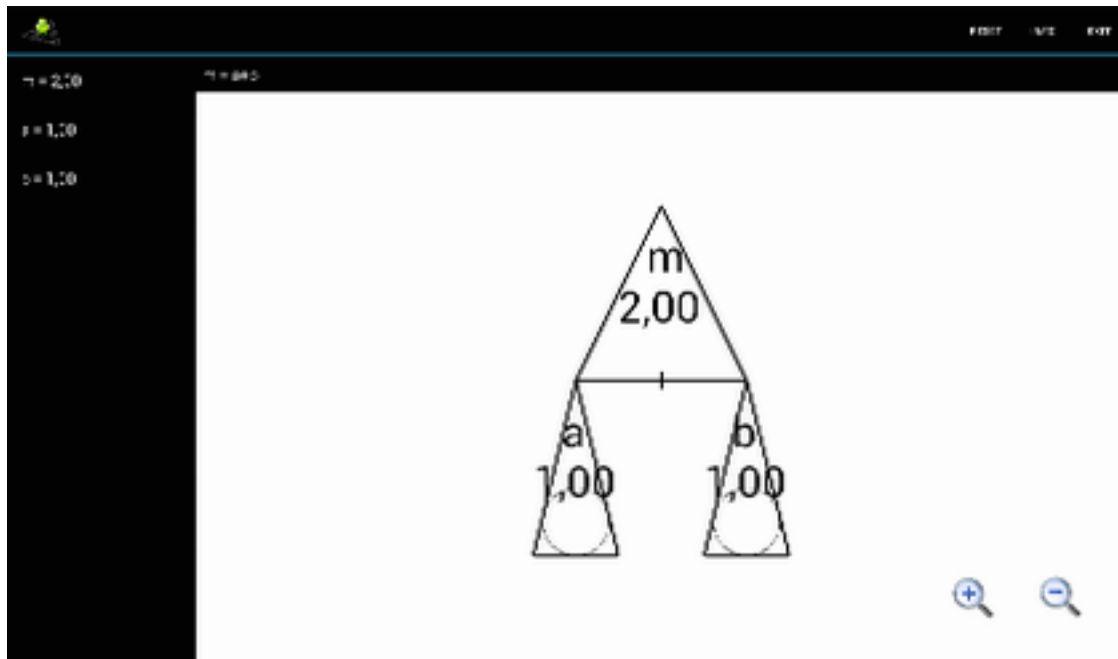


Figura 26: Interfície definitiva de l'aplicació

Es pot comprovar com la interfície ha respectat el disseny del mockup. Cal comentar que a la barra superior s'han afegit tres opcions necessaries per completar l'aplicació.

1. Reset: Amb aquest botó es neteja tot l'arbre així com la fórmula i la llista de manera que es torna a l'estat inicial on només hi ha un node amb valor 2. S'ha afegit per tal de que no s'hagi de sortir i tornar a entrar de l'aplicació.
2. Info: Clickant sobre aquesta opció s'obre un dialog amb informació sobre el desenvolupador de l'aplicació i l'any de creació.
3. Exit: Amb aquesta opció se surt del programa

Resultats

En aquest punt s'ha finalitzat el període destinat al projecte. S'ha pogut sotmetre a test i s'han extret conclusions per poder-lo depurar i per poder treure versions més completes en un futur. L'avaluació del producte s'ha fet durant les diferents iteracions del projecte per part dels membres del grup de desenvolupament i utilitzant a terceres persones fent tests d'usabilitat més o menys guiats.

Les proves s'han fet en entorns controlats provant cada una de les funcionalitats de l'aplicació però també s'han portat a terme jornades en les que s'han proposat exemples a les persones del públic per tal de poder avaluar com de fàcil és comprendre el funcionament de l'aplicació [Anex A i Anex B].

Els resultats d'avaluar aquesta aplicació no poden ser quantitatius, per tant s'han anat recopilant opinions de la gent avaluada i els resultats han sigut satisfactoris, demostrant que s'han assolit els objectius tot i que hi ha aspectes millorables.

Les avaluacions han demostrat que l'aplicació causa bona impressió i resulta intuïtiva i agradable a la vista, tot i això, a la gent no familiaritzada amb els dispositius tàctils els hi costa aquest tipus d'interacció. De totes maneres la complexitat de l'aplicació radica en el coneixement de les diferents àrees clickables i dels gestos que s'han de fer per cada acció. Amb els tests fets es pot afirmar que amb una sessió de dues hores amb un públic familiar es pot dominar l'aplicació perfectament.

Paràmetres

A l'aplicació, al tenir zones concretes clickables molt properes s'han hagut de marcar uns llinars per definir les mides d'aquestes zones. Aquests llinars s'han anat ajustant segons les proves d'usabilitat i finalment s'han definit com mostra la [figura X]. Definint l'alçada del triangle com h , s'ha determinat que per considerar que l'usuari ha clickat sobre la base del triangle, s'ha de trobar a una distància paral·lela inferior a $0.15 \cdot h$. De la mateixa manera, per considerar que l'usuari ha clickat sobre el vèrtex superior del triangle, es considera que $0.15 \cdot h$ és un radi prou ample per encertar sense preocupacions i prou estret com per no solapar-se massa amb àrees d'altres triangles. La resta d'àrea del triangle es considera centre.

Recordem que fem coincidir el vèrtex superior de tot triangle, exceptuant el node arrel, amb la base del node pare, per tant s'ha d'establir un sistema de preferències. Considerant que la base del triangle és prou ample, o si més no es pot fer prou ample variant l'escala horitzontal, com per tenir espai suficient com per no haver de tocar un vèrtex superior d'un node fill, es dóna prioritat als vèrtexs superiors per sobre de les bases. Un segon conflicte el trobem amb les línies divisòries que es troben a la base del triangle. Donat que aquestes línies tenen una funció redundant, ja que modificant directament l'àrea d'un node es pot arribar al mateix resultat, també donarem prioritat en aquest cas al vèrtex superior per sobre de les línies divisòries. L'àrea disponible per clicar sobre aquestes línies és de 20 píxels de radi.

Exemple pràctic

Un cop tenim l'explicació de l'aplicació acabada, ve el moment d'un cas pràctic per veure les possibilitats d'aquest software. En el següent exemple s'il·lustrarà i s'explicaran els passos a seguir per representar una fórmula matemàtica convencional i es mostrarà una imatge de cada pas. La fórmula escollida forma part de l'àmbit mèdic, per a pacients amb problemes de calci. La fórmula és: Correcció de calci sèric total (mg/dl) = $\text{Calci total (mg/dl)} + 0.8 * (4 - \text{Albumina(g)})$. Aquesta fórmula es resumirà com $\text{Ca corrected} = \text{Ca pl} + 0.8 * (4 - \text{Albumina})$

Els passos a seguir són els següents:

1. Obrir l'aplicació, veurem que ja es troba el primer node.

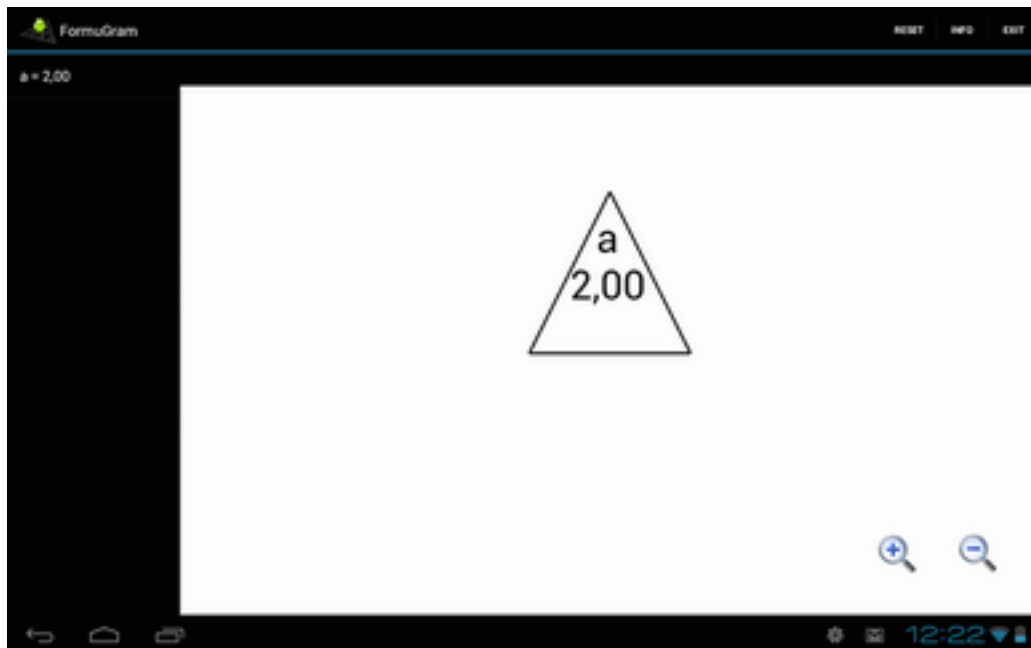


Figura 26: Pas 1.

2. Crear dos nodes fills del primer.

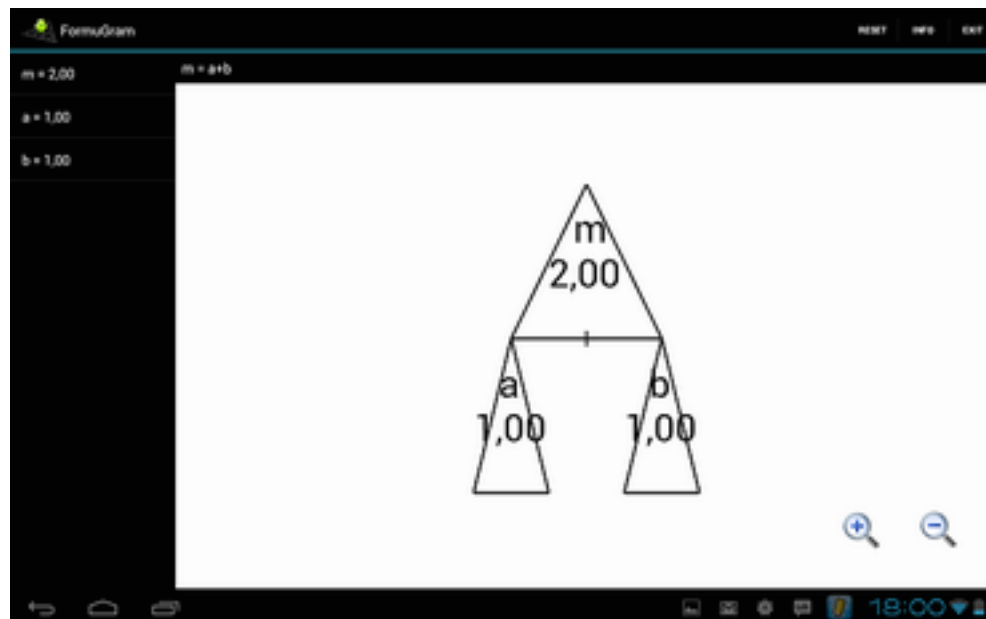


Figura 27: Pas 2

3. Creem dos fills del fill dret. Per fer això, es bloquejarà el fill esquerre i es posarà el fill dret a 2.00 per una millor visualització

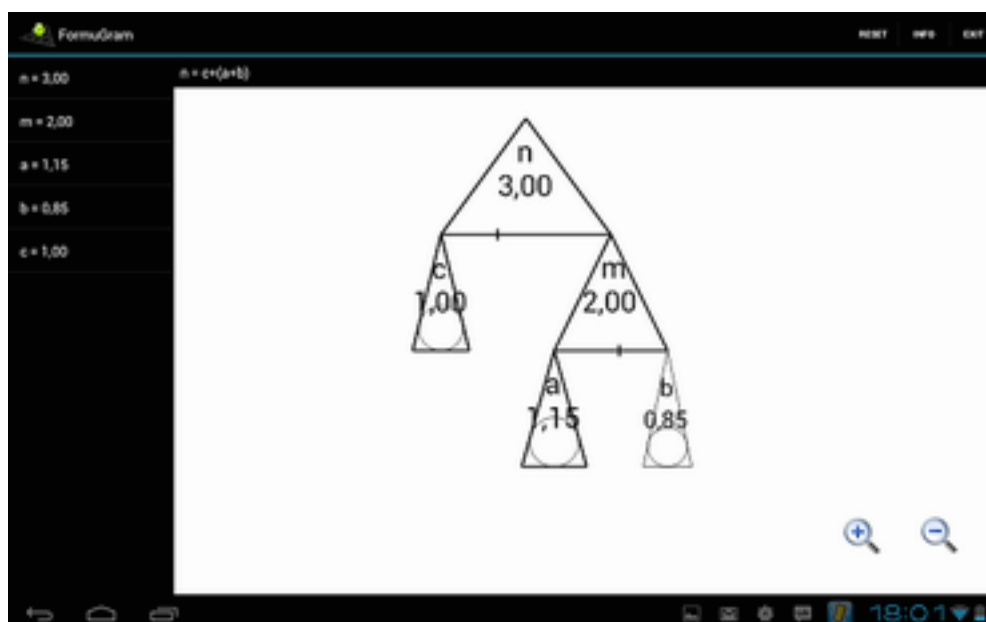


Figura 28: Pas 3

4. Assignem la constant 0.8. S'ha de modificar l'àrea del triangle 'a'.

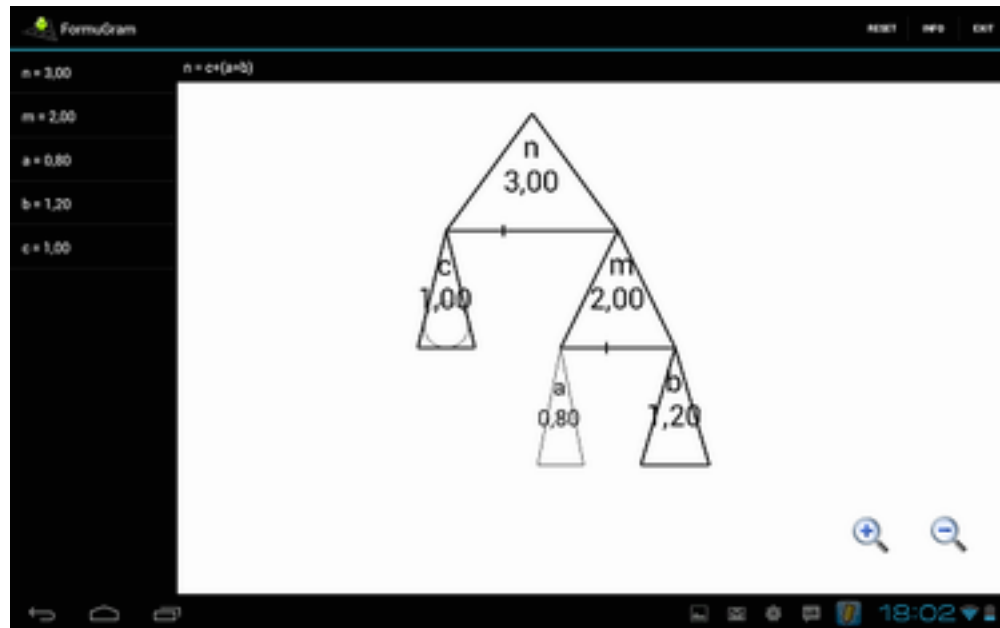


Figura 29: Pas 4

5. Fem gran el node 'b' i creem dos nodes fills seus. En aquest moment tindrem tots els nodes de l'arbre necessaris i es podrà començar a jugar amb els valors.

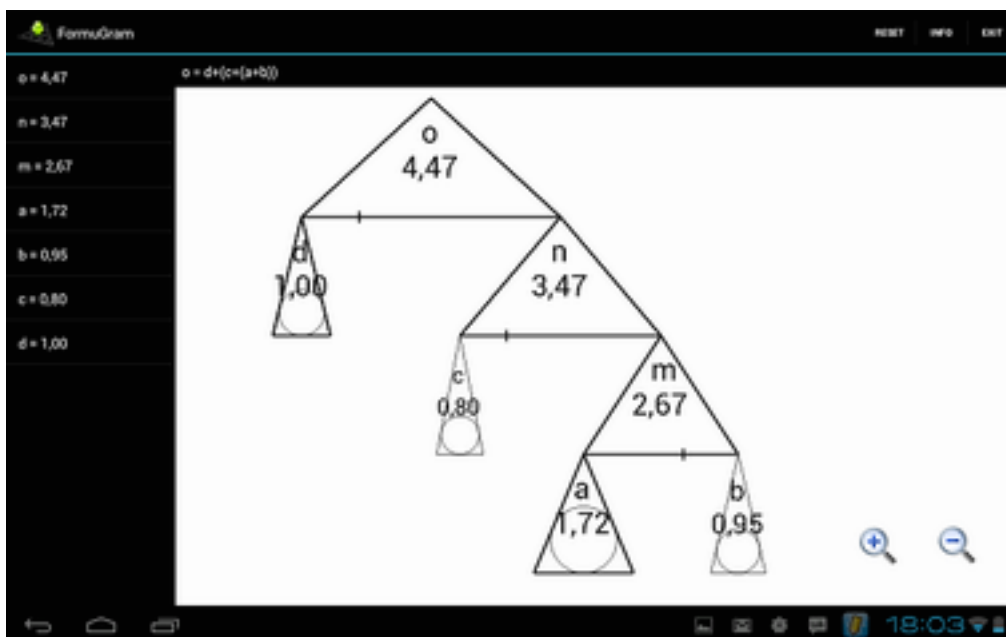


Figura 30: Pas 5

6. Modifiquem el node 'a' per crear la constant 4.

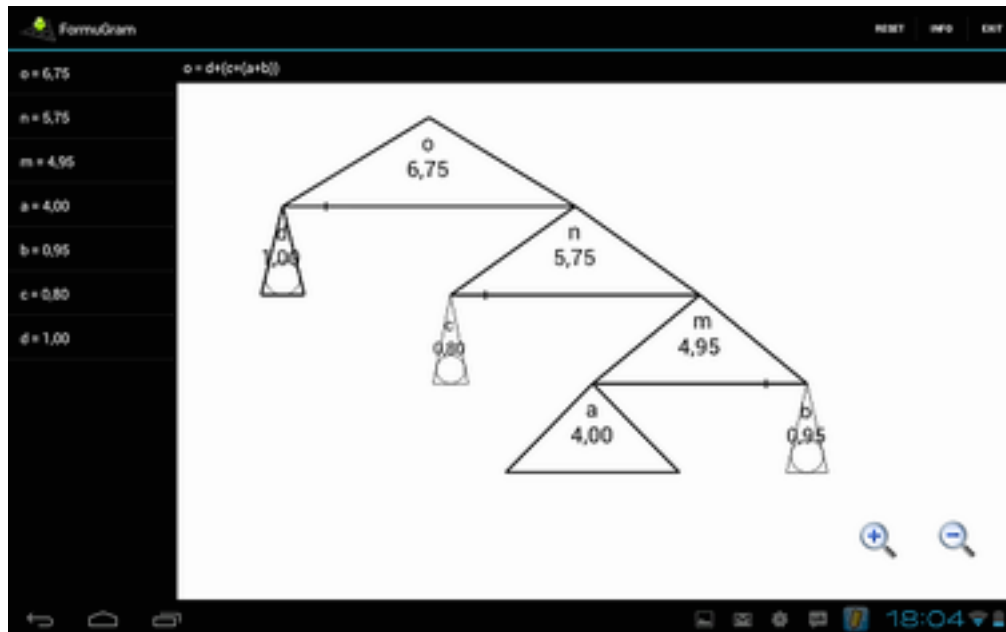


Figura 31: Pas 6

7. Canviem la operació del node 'n' perquè passi a multiplicar 'c' i m'.

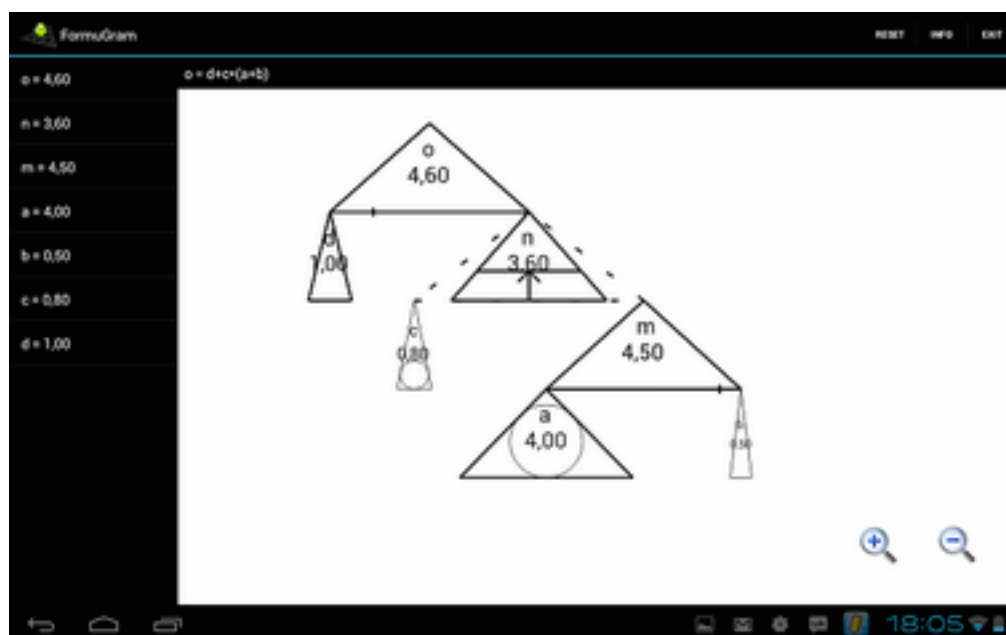


Figura 32: Pas 7

8. Es posen els noms a les variables per tenir una millor referència del que representen.

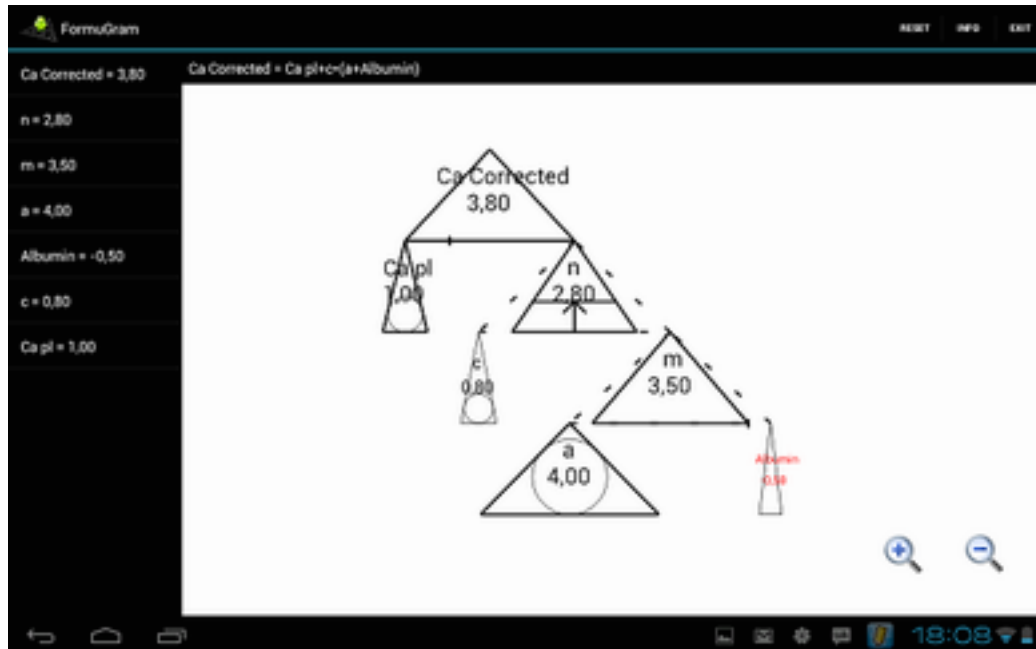


Figura 33: Pas 8

Amb aquest exemple hem pogut comprovar la simplicitat amb la que es pot representar una fórmula matemàtica amb l'Equdia. A partir d'aquest moment, modificant el valor de les variables de l'equació, deixant sempre una com a entrada i una com a sortida, podem observar com reacciona l'arbre en temps real.

Conclusions

Després de tot aquest temps arriba el moment de fer les valoracions de tot el treball. Els objectius marcats inicialment han sigut assolits satisfactòriament ja que s'han implementat tots els casos d'ús, l'aplicació funciona sense errors i s'ha pogut testear i divulgar com era desitjat.

De totes les probes fetes es pot extreure que el sector on l'Equdia té millor acceptació és entre els nens de primària. És un públic amic de les noves tecnologies i per a molts suposa un joc transformar equacions en gràfiques i poder jugar amb elles, per molt bàsiques que aquestes puguin ser.

En quant a les possibles millores que es podrien implementar per futures versions de l'aplicació és necessari comentar petits problemes de plantejament que s'han trobat i millores que s'han estat plantejant:

- 1 - Un disseny menys sobri. La interfície té un disseny massa antiquat, podria ser més amigable, sobretot pel fet que està enfocat a la docència dels més petits.
- 2 - Noves funcions. Es poden afegir noves funcions a les existents perquè ara per ara el nombre d'equacions que es poden representar és molt limitat.
- 3 - Poder enllaçar variables. En la majoria de fórmules matemàtiques una variable pot aparèixer més d'un cop, en canvi actualment l'aplicació, tot i que es pot posar el mateix nom a dues variables i així simular aquest aspecte, harien de poder estar enllaçades de tal manera que en modificar una, l'altre ho fes de la mateixa manera.

Personalment, com a desenvolupador i tester de l'aplicació, trobo que el major inconvenient és el gran nombre d'interaccions semblants i sobre els mateixos punts que hi ha, ja que tocar zones com el vèrtex superior del triangle pot suposar modificar el valor d'una variable, limitar el seu rang i variar l'escala horitzontal. Això és un punt crític que pot provocar molts errors. Per tant abans de posar el dit a qualsevol lloc s'ha de tenir molt clar el que es vol fer.

En general em sembla satisfet amb el treball fet ja que he complert amb les expectatives i m'ha agradat participar en jornades de divulgació on he pogut mostrar al públic el meu projecte.

Referències

- [1] GESTIÓ DE PROJECTES Programació i context gràfic, per Josep Tarrés i Turon.
- [2] http://ca.wikipedia.org/wiki/Triangle_de_Sierpi%C5%84ski
- [3] http://ca.wikipedia.org/wiki/Triangle_de_Tartaglia
- [4] <http://www.tucanet.com/2012/07/05/buscadores-sistemas-operativos-y-navegadores-web-m%C3%A1s-populares-a-nivel-mundial/>
- [5] <http://androideity.com/2012/07/16/5-lenguajes-para-programar-en-android/>
- [6] <http://www.droiddraw.org/>
- [7] <http://appinventor.mit.edu/>
- [8] http://www.mono-project.com/Main_Page
- [9] <http://www.basic4ppc.com/>
- [10] <http://www.runrev.com/>
- [11] <http://developer.android.com/tools/sdk/eclipse-adt.html>
- [12] <http://www.gencat.cat/salut/ticsalut/html/ca/dir1738/doc13424.html>
- [13] <http://www.parcudg.net/jornadaticsalut/>
- [14] http://www.fundaciorecerca.cat/Agenicies.asp?accion=VerGrande&IdAgenicies=383#.UPIESR2-r_A

Anex A: TicSalut

La tercera edició de les Jornades R+D+I en TIC i Salut es va dur a terme els dies 7 i 8 de juny del 2012 i va comptar amb 250 assistents, la participació de 26 grups i centres de recerca, 42 empreses i 7 centres assistencials i s'hi van presentar una vuitantena de projectes innovadors. En aquesta edició es va presentar l'Equidia per primera vegada de manera pública.

Missió

TicSalut té la missió d'assolir la plena incorporació del sistema sanitari català a la societat del coneixement.

Objectiu general

Impulsar el desenvolupament i la utilització de les tecnologies de la informació i la comunicació (TIC) i el treball en xarxa en l'àmbit de la salut, avançant cap a un model basat en l'atenció personalitzada i humana per a tota la ciutadania.

Objectius específics

1. Elaborar criteris, orientacions estratègiques i informació útil per a la presa de decisions sobre tecnologies de la informació i la comunicació en l'àmbit de la salut.
2. Oferir suport tècnic a l'Administració, els usuaris i els proveïdors de solucions tecnològiques.
3. Introduir aplicacions tecnològiques capdavanteres en l'àmbit de salut que facilitin la transferència ràpida de coneixements entre els centres de salut i els professionals, permetent la millora dels serveis sanitaris dirigits a ciutadans i ciutadanes.
4. Promoure el moviment d'informació abans que el moviment de persones, assolint el més alt grau d'eficiència en el diagnòstic.
5. Esdevenir un centre de referència a Catalunya, treballant en xarxa amb altres entitats d'Espanya, Europa i d'arreu del món.

Anex B: Festa de la Ciència

El cap de setmana del 16 i 17 de juny al Parc de la Ciutadella de Barcelona es va celebrar la 6a FESTA DE LA CIÈNCIA. L'objectiu principal de la FESTA enguany va ser fer èmfasi en la recerca i la innovació d'alt nivell que es fa al nostre país convidant no només a les institucions, universitats i centres de recerca sinó també a les empreses i els seus departaments d'R+D+I a mostrar el potencial de creació de coneixement com a motor de desenvolupament.

Aquest 2012 ha destacat el protagonisme de la tecnologia, no només amb un espai on es combinen instal·lacions, espectacles i jocs amb connexió remota sinó també com a fil conductor entre els diferents punts d'activitat. També s'ha centrat bona part de les activitats a les energies sostenibles i la neurociència.

El model de festa consolidat se centra en espais on se succeeixen propostes diverses, liderades per investigadors i divulgadors de diferents centres, que posen en valor la capacitat de fer-se preguntes i buscar respostes a partir del raonament i l'experimentació. L'horari, que cobreix dissabte tarda i nit i el matí de diumenge, permet la combinació d'activitats a l'aire lliure, com els itineraris, els jocs o l'observació astronòmica, amb la projecció audiovisual o les instal·lacions més espectaculars en la franja nocturna. L'atenció a públics d'edats i interessos diferents es reflecteix en la combinació de formats: des d'espectacles, demostracions, jocs i un espai de lectura per als més petits fins a tallers i instal·lacions per a joves i adults.

FORMUGRAM

Representació Gràfica de Fórmules
Oriol Janés, Josep Tarrés, Oriol Pujol, Sergio Escalera

1

j

Representació clàssica de fòrmules

Exemple: Fòrmula de De Moivre

$$\cos nz = \frac{(\cos z + i \sin z)^n + (\cos z - i \sin z)^n}{2}$$

Inconvenients:

- Comprensió poc intuïtiva
- Interacció lenta i costosa

► 2

Alternativa: Representació geomètrica

p.e.: Representació de jerarquies

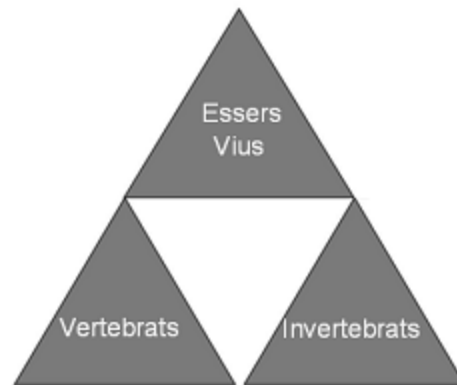


Comprensivament intuïtiva, basada en la posició dels objectes.

► 3

Alternativa: Representació geomètrica

p.e.: Representació de jerarquies

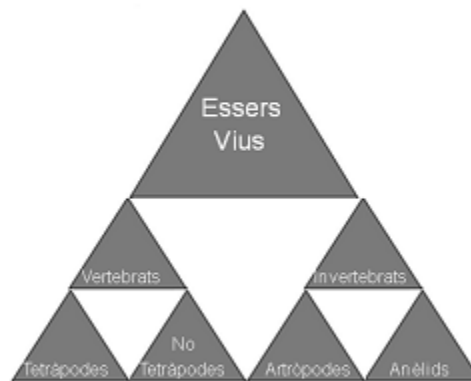


Comprensivament intuïtiva, basada en la posició dels objectes.

► 4

Alternativa: Representació geomètrica

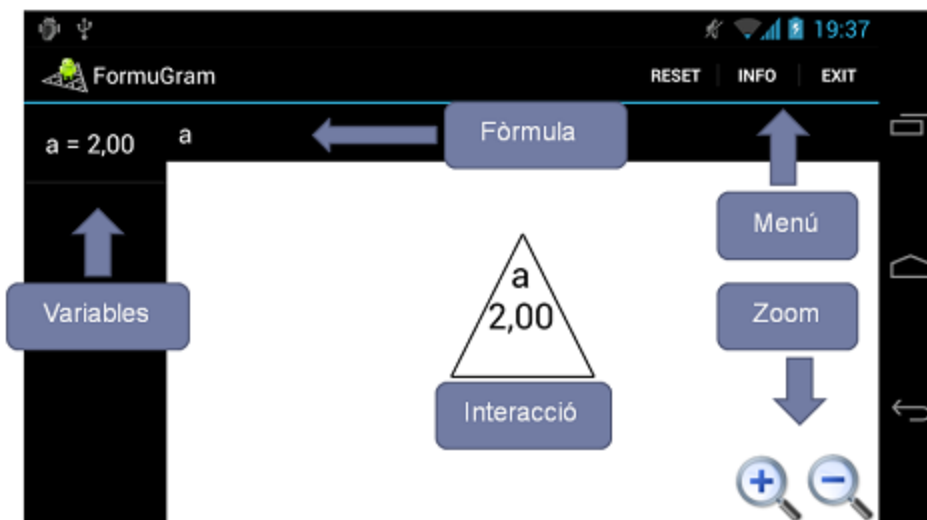
p.e.: Representació de jerarquies



Comprensivament intuïtiva, basada en la posició dels objectes.

► 5

Introducció al FormuGram



▶ 6

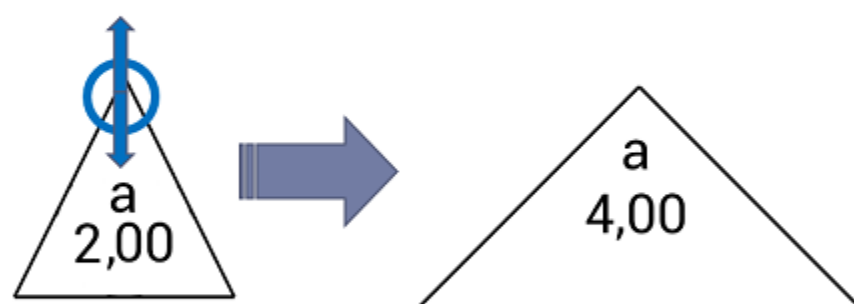
Exemple

- ▶ Un zoològic vol fer càlculs per dividir la seva extensió en diferents àrees.
- ▶ L'àrea consta de quatre hectàrees i està dividida en animals **aquàtics i animals terrestres**. També es vol dividir el territori dels terrestres entre **rèptils i mamífers**.

▶ 7

Interaccions

- ▶ Variació del valor d'una variable (triangle)

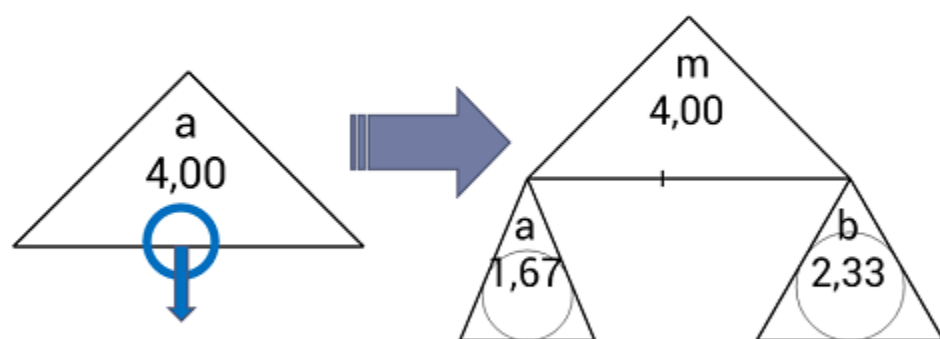


Clickar sobre l'extrem superior del triangle i moure verticalment

▶ 8

Interaccions

- ▶ Creació de fills



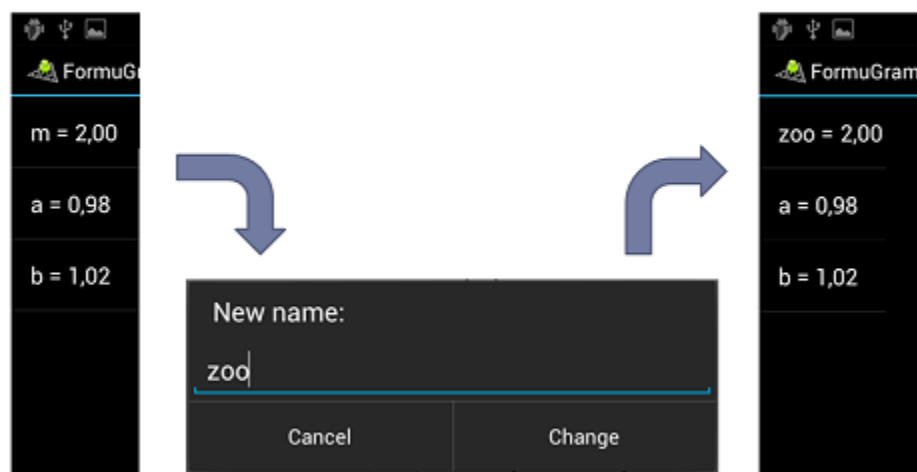
Moure cap avall des de la base del triangle

La mida dels triangles resultants depèn del punt exacte premut

▶ 9

Interaccions

► Renombrar les variables

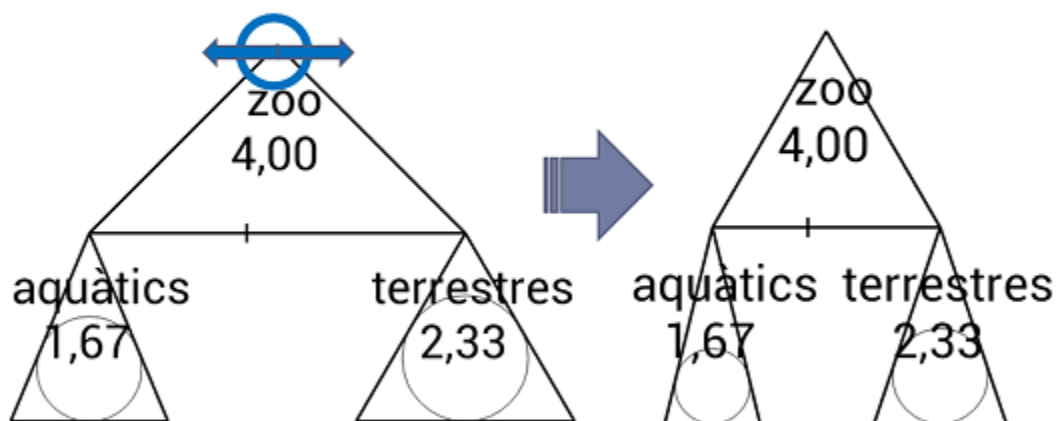


Clickar sobre una variable per canviar el nom

► 10

Interaccions

► Canvi d'amplada

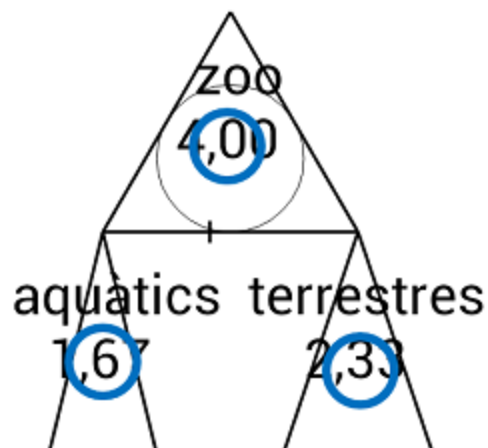


Clickar sobre l'extrem superior i moure el dit horitzontalment

► 11

Interaccions

- ▶ Bloqueig / Desbloqueig

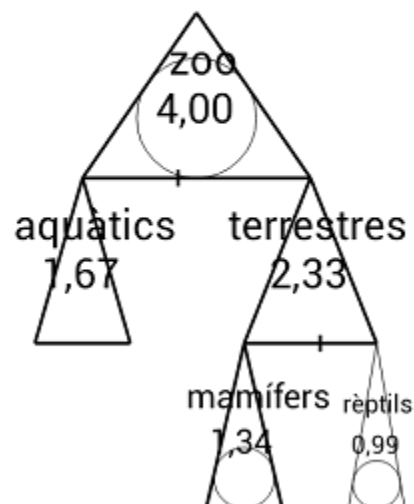


Clickar al centre del triangle

▶ 12

Interaccions

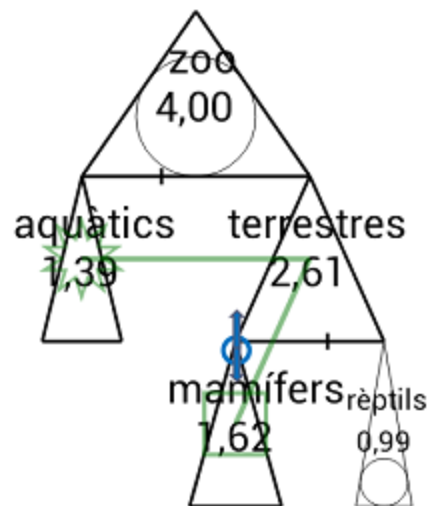
- ▶ Finalitzant els preparatius



▶ 13

Interaccions

- ▶ Entenent els camins i fent els càlculs



▶ 14

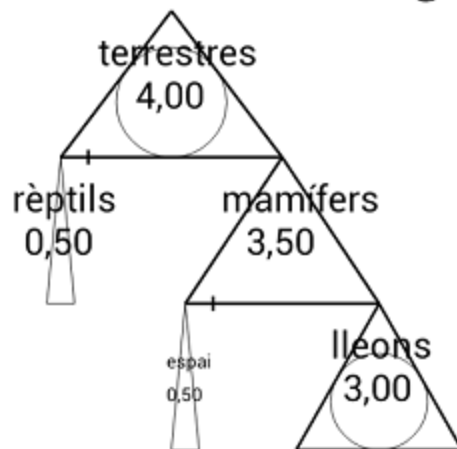
Exemple 2. Introducció a la multiplicació

- ▶ Un cop arribats a aquest punt, el zoològic ha fet canvis i volen distribuir l'espai un altre cop.
- ▶ Aquest cop només tenen rèptils, i a la zona de mamífers, tenen tres lleons que necessiten molt espai. Quant espai tindrà cada lleó?
- ▶ La modificació del gràfic es farà en tres passos:
 - Renomenar les variables
 - Assignar la constant 2.00
 - Convertir la operació, de suma a multiplicació

▶ 15

Exemple 2. Introducció a la multiplicació

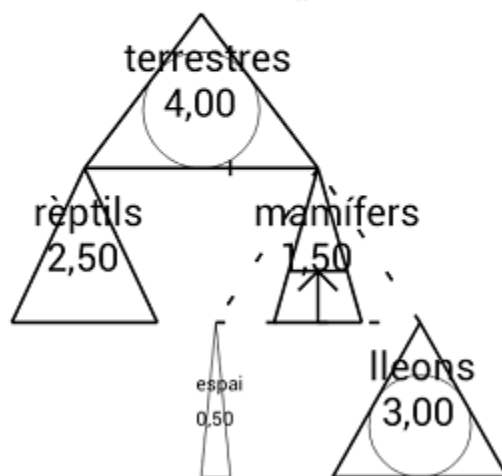
- ▶ Abans d'aplicar la multiplicació preparem les dades.
- ▶ El diagrama hauria d'estar de la següent manera.



▶ 16

Exemple 2. Introducció a la multiplicació

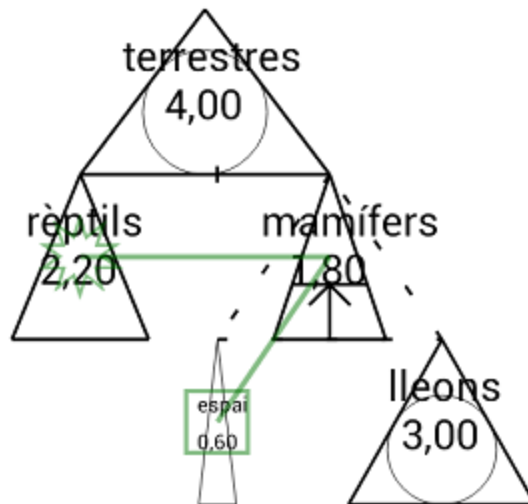
- ▶ Al deixar premut el centre del triàngle 'mamífers', la operació canvia a multiplicació.



▶ 17

Exemple 2. Introducció a la multiplicació

- ▶ Ara, al modificar els valors, veiem com es manté la proporció entre les variables.



▶ 18

Gràcies per la seva assistència

▶ 19