

# CkEditor Plugin per generar mapes d'imatges

## Memòria del projecte

**Universitat Politècnica de Catalunya**

**Facultat d'Informàtica de Barcelona**

**Especialitat d'Enginyeria del Software**

**Autor:** Joan Torras Castillo

**Director:** Roger Espasa Sans

## Índex

1. Context del projecte.....	7
1.1 Problema .....	7
1.2 Estat de l'art .....	8
1.3 Analitzem la competència.....	9
EDMODO .....	9
SOCRATIVE.....	9
QUIZLET.....	10
Conclusions.....	10
2. Actors implicats (stakeholders).....	11
2.1 A qui va dirigit .....	11
2.2 Altres actors .....	12
3. Abast del projecte .....	13
3.1 Plugin de mapeig d'imatges .....	13
3.2 Activitat interactiva .....	14
4. Planificació temporal.....	16
4.1 Duració del projecte .....	16
4.2 Divisió del projecte.....	16
Planificació del projecte:.....	17
Fase Inicial: .....	17
Primera fase de desenvolupament:.....	18
Segona fase de desenvolupament: .....	19
Tercera fase de desenvolupament:.....	20
Fase final:.....	21
Hores totals: .....	21
4.3 Valoració d'alternatives i pla d'acció.....	22
4.4 Recursos .....	22
5. Gestió econòmica.....	23
5.1 Identificació dels costos .....	23
5.2 Costos humans .....	23
5.3 Costos materials i Software.....	23
5.4 Possibles imprevistos .....	23
5.5 Costos totals.....	24
6. Solució proposada .....	25

7. Per què CkEditor?.....	29
8. Planificació del projecte .....	31
8.5 Diagrama d'estats.....	39
9. Implementació .....	42
9.1 Paràmetres de configuració de CkEditor.....	42
9.2 Generació del plugin QuizEditor .....	43
9.3 Creació de la pantalla d'edició d'activitats.....	45
9.4 Manager.js.....	47
9.4.2 Càrrega de l'editor.....	48
9.4.5 Funcions per guardar i carregar dades.....	59
9.5 Altres classes del projecte .....	60
9.8 Generació de l'apartat d'execució d'activitats.....	61
10. Plugins complementaris a QuizEditor .....	65
10.3 Altres plugins d'edició d'activitats .....	65
11. Sostenibilitat i compromís social.....	66
11.1 Ambiental .....	66
11.2 Econòmic .....	66
11.3 Social .....	66
11.4 Matriu de sostenibilitat.....	67
12. Conclusions.....	68
13. Bibliografia .....	69

## Resum

En aquest projecte s'ha desenvolupat una eina per a la docència que permeti crear una activitat amb un format de pregunta-resposta amb correcció automàtica, especialment enfocada a preguntar de manera interactiva sobre material visual (mapes, fotografies, diagrames).

Aquesta eina estarà incorporada en un *plugin* dins un editor de text en línia, com és el *CkEditor*, que serveixi pel professorat com a eina per crear activitats interactives per als alumnes mitjançant imatges. El professor podrà inserir mapes dins les imatges i afegir-hi quadres de text que contindran preguntes i apareixeran quan l'alumne cliqui sobre ells.

La intenció del projecte és que aquesta eina sigui fàcilment usable per qualsevol usuari, que ofereixi al professor moltes possibilitats per personalitzar l'activitat que entregará a l'alumne i que aquesta sigui adaptable a diversos àmbits d'estudi.

## Resumen

En este proyecto se ha desarrollado una herramienta para la docencia que permite crear una actividad con un formato de pregunta-respuesta con corrección automática, especialmente enfocada a preguntar de forma interactiva sobre material visual (mapas, fotografías, diagramas).

Esta herramienta estará incorporada en un *plugin* dentro de un editor de texto en línea, como es *CkEditor*, que sirva al profesorado como herramienta para crear actividades interactivas a sus alumnos mediante imágenes. El profesor podrá insertar mapas dentro de las imágenes e introducir cuadros de texto que contendrán preguntas i aparecerán cuando el alumno clique sobre ellos.

La intención del proyecto es que esta herramienta sea fácilmente usable para cualquier usuario, que ofrezca al profesor muchas posibilidades para personalizar la actividad que entregará al alumno y que sea adaptable a diversos temas de estudio.

## **Abstract**

In this Project has been developed a teaching's tool which allows to create an activity with an automatic correction question-answer format, specially focused on ask interactively about visual's material (maps, photos, diagrams).

This tool will be incorporated in a plugin inside an online text editor, like *CkEditor* is, which professorship could use it like a tool to create interactive activities to their students using images. Professor could insert maps inside the images and introduce text boxes which will contain questions and will appear when students clicks on them.

The project's intention is to make this tool easily usable to any user, and offer to the professorship an amount of possibilities to personalize the activity that the student will receive and adapt it to many study subjects.

# 1. Context del projecte

## 1.1 Problema

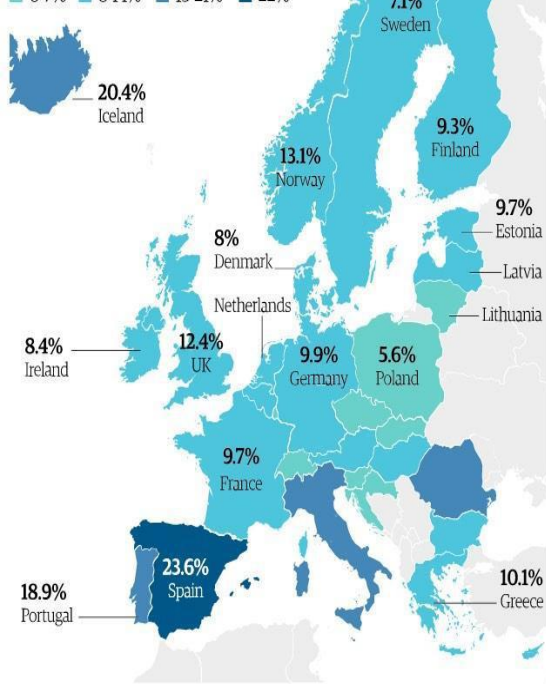
L'any 2015 la taxa de fracàs escolar a Espanya després de cursar l'educació secundària va arribar pràcticament al 20% segons un estudi realitzat per la *Encuesta de la Población Activa* (EPA). Aquesta xifra destaca molt si la comparem amb la mitja Europea que no arriba al 10%[1].

Es poden trobar molts motius pels quals arribem a aquestes xifres tan elevades, però tots acabant apuntant a la falta de motivació dels alumnes per aprendre, que és conseqüència de la falta d'eines de les quals disposen els professors per estimular a qui ensenyen. Per aquesta raó veiem que a les escoles privades la taxa de fracàs escolar és més reduïda que a les escoles públiques, ja que aquestes disposen de més recursos per augmentar les ganes d'aprendre de l'alumne[2].

### School dropout rates in the EU

18-24 year-olds, Percentage

0-7% 8-14% 15-21% 22%>



GUARDIAN GRAPHIC

SOURCE: EUROSTAT

Sabent això podem veure que una de les raons més importants per la desigualtat del fracàs escolar a Espanya en comparació amb altres països europeus, és l'escassa inversió que fa l'estat en l'educació. En general, si s'augmenten els recursos dels professors a les aules, augmenta també les possibilitats que aquests tenen per estimular als seus alumnes

Per això és important que apareguin eines que puguin ajudar al professorat sense suposar un cost molt elevat. A l'hora d'invertir en educació, les escoles ja estan optant per les noves tecnologies, ja sigui amb l'ús de projectors, incorporant ordinadors a les aules o amb la creació de portals web (*moodle*) que afavoreixin la comunicació entre el professorat i l'alumne entre d'altres.

Aquest projecte oferirà una eina senzilla dins un editor de text en línia on el professorat podrà proposar activitats als seus alumnes utilitzant imatges i podrà preguntar conceptes d'una manera interactiva. Els alumnes tindran la possibilitat de realitzar

l'activitat a través d'internet i rebre al moment el feedback si la seva resposta és correcta o no.

## 1.2 Estat de l'art

Els pedagogs han estat utilitzant moltes tècniques des de fa molts anys per captar l'atenció i estimular l'alumnat amb la intenció d'augmentar les seves ganes d'aprendre. Però aquesta no es una tasca senzilla, i és cada vegada més complicada amb els escassos recursos dels que disposen els centres. Moltes vegades la raó per la qual els alumnes perden la motivació per l'escola es per que senten que no estan aprenen el que deuriem amb tot el temps que passen a l'escola.

Degut al sistema organitzatiu actual de la docència en classes massificades, l'alumne en moltes ocasions es pot arribar a sentir perdut a l'hora d'enfrontar-se amb una activitat proposada per el seu professor. Materialment, els docents no tenen el temps necessari per proporcionar el seguiment immediat i personalitzat que pot necessitar un alumne en particular. Si comparem el sistema docent actual amb la funció que realitzava un pedagog a l'antiga Grècia, que es dedicava a tutoritzar un alumne individualment, trobem que en l'actualitat l'alumne no rep el *feedback* que necessita quan està realitzant una activat.

Troben molts estudis que mostren la importància d'un *feedback* immediat per augmentar la facilitat amb que un alumne aprèn conceptes[3][4]. I és aquest el repte que es troben molts docents al seu dia a dia i no tenen les eines suficients per afrontar-ho.

Les tecnologies de la informació poden ajudar el docent a proporcionar feedback immediat fins i tot en grups de molts alumnes fent la part de la correcció de les activitats de manera automàtica i interactiva.

Però aquesta no es la única avantatge que ens poden aportar les tecnologies de la informació, també tenen un paper important a l'hora de fer més atractiu per l'alumne la manera d'aprendre. Una de les raons per la que això es així es gràcies al poder de les il·lustracions. Aquests sempre han sigut un element essencial dins els llibres de text, ja que per una banda son un element decoratiu que fa que l'alumne vegi menys farragós el text que l'acompanya i per l'altre serveixen per descriure conceptes d'una manera més senzilla per l'estudiant[5].

Per aquesta raó molts professors ja estan optant per proposar als alumnes activitats a través de la web i des de fa molts anys estan sorgint plataformes per que el professorat tingui una major facilitat per poder posar activitats als alumnes de la forma més estimulante possible.

El projecte que es proposa en aquest document pretén atacar les mancances del sistema organitzatiu actual mitjançant un eina per que el professorat pugui crear, de forma senzilla, activitats interactives de correcció automàtica per l'alumne, utilitzant el mètode de pregunta-resposta adjuntat a un contingut visual.



Es cert que ja es troben al mercat moltes eines que tenen una funció semblant en diversos camps, però mai s'acaben d'ajustar al que necessita un docent en un moment concret. És per aquesta raó que hi donem molta importància a que l'eina que es proposa en aquest document sigui fàcil d'usar i adaptable a tots els entorns.

### **1.3 Analitzem la competència...**

En aquest apartat analitzarem tres eines populars semblants a la que es proposa en aquest document per saber quines bones pràctiques deviem seguir i quines no.

#### EDMODO

Edmodo és una pàgina web educativa plantejada com una xarxa social per professors, pares i alumnes. Amb aquesta pàgina els professors poden comunicar-se amb els alumnes, crear avisos per a ells, compartir documents, fer proves i realitzar activitats interactives o utilitzar les creades per altres usuaris.

Edmodo és una eina molt útil que dona moltes possibilitats als professors per crear activitats personalitzades pels estudiants. Però el seu principal problema és la gran complexitat que es poden trobar professors i alumnes a l'utilitzar-la. Encara que és una pàgina amb molt de futur no sembla molt usable per professors que no dominen les eines TIC i alumnes d'edats reduïdes.

#### SOCRATIVE

Socrative és una pàgina web per professors i alumnes orientada a la realització de proves pels alumnes, i mantenir un control de les seves respostes. Amb aquesta eina els professors poden crear proves interactives de tipus pregunta-resposta i veure en temps real les respostes dels alumnes. Per altra banda els alumnes reben un feedback quan finalitzen cada prova.

Socrative és una eina senzilla i útil perquè els professors creïn proves pels alumnes. Està plantejada per alumnes d'edat reduïda, ja que és fàcil d'utilitzar, només cal un senzill registre, i té una interfície acolorida i estimulante. És una eina molt orientada a què el professor pugui tenir un seguiment sobre un alumne concret, però les opcions del professor estan molt limitades a realitzar els qüestionaris típics amb una interfície gràfica cridanera.

## QUIZLET

Quizlet és una pàgina web on els docents poden crear unitats d'estudi per als alumnes. En aquestes unitats introdueixen una documentació del tema a tractar i uns termes que els alumnes han d'aprendre. Totes les unitats tenen unes activitats definides que relacionen els termes amb les seves definicions mitjançant proves interactives o jocs. Els usuaris de la pàgina, sense necessitat d'un registre previ poden realitzar qualsevol activitat que hagi creat un altre usuari.

És una eina senzilla pels professors, té una interfície intuïtiva i dóna un bon feedback a l'estudiant. El seu principal problema és que totes les unitats docents tenen les mateixes proves amb els termes introduïts pel professor, i en molts casos moltes de les proves manquen de sentit i es tornen avorrides. Encara que la idea de la pàgina és bona per com està plantejada no sembla molt útil en la majoria d'àmbits d'estudi.

## Conclusions

Observant els diferents tipus d'eines que existeixen al mercat veiem la dificultat que hi ha per crear una eina senzilla adaptable a àmbits variats. És per això que el projecte que es proposa en aquest document se centrarà en la realització d'una interfície senzilla pel professorat, però que a la vegada doni joc a què es pugui personalitzar l'activitat que es vol per tal de ser usable en un ampli ventall d'àmbits d'estudi.

## 2. Actors implicats (*stakeholders*)

Per fer una correcta realització del projecte que es presenta en aquest document és essencial estudiar quins seran els actors que interactuaran amb ell. Cal saber quin és el públic a qui va dirigit aquest projecte i es beneficiarà d'ell i quins són els actors que tindran una certa implicació.

### 2.1 A qui va dirigit

#### Alumnes

Els alumnes seran els actors principals d'aquest projecte, ja que són ells els que han de resoldre els problemes plantejats pels professors i és a ells a qui va dirigit aquest projecte. Tenint en compte que els alumnes que hauran d'actuar amb la nostra aplicació poden tenir una edat reduïda cal plantejar la interacció amb l'aplicació de manera que els hi sembli el més estimulant i divertida possible.

Però cal tenir en compte també que aquesta eina es podria utilitzar també en àmbits més adults, així que encara que l'enfoquem per al jovent ha de ser suficientment neutre com perquè es pugui utilitzar, per exemple, en l'àmbit universitari si s'escau.

#### Professors

Els professors són els actors complementaris del projecte, ja que són ells qui utilitzaran l'eina que estem plantejant i crearan el material per als alumnes. És evident que la creativitat dels professors serà determinant per aconseguir l'engrescament dels estudiants.

Hem de pensar que no tots els professors són experts en eines TIC i per tant la interfície ha de ser suficientment intuïtiva sense necessitat de llegir-se una documentació excessiva.

#### Pares i altres consumidors

Encara que és raonable pensar que majoritàriament serà usada per professors, certament l'eina està completament oberta a pares i educadors en general. Pot ser una bona opció per proposar activitats per part de les famílies que serveixin per reforçar el que aprenen a classe un estudiant.

## 2.2 Altres actors

### Escoles

Les escoles són un actor important pel projecte, ja que són elles les que poden divulgar el projecte al professorat perquè l'utilitzin. També és cert que les escoles poden ser beneficiaris utilitzant el projecte perquè gràcies a ell es podria augmentar la dedicació dels alumnes i afectar de manera positiva les seves qualificacions.

### Desenvolupadors d'activitats

Aquest projecte està orientat a la realització d'activitats interactives, i existeixen en el mercat moltes empreses dedicades a crear editors d'activitats interactives útils per poder oferir bons exercicis a l'alumnat de manera interactiva utilitzant les noves tecnologies. Algunes d'aquestes empreses utilitzen també editors de text en línia per realitzar aquestes activitats, i són aquestes, les que gràcies al plugin que es desenvolupa en aquest projecte tindran a la seva disposició una nova eina a oferir als docents que utilitzin les seves aplicacions.

### CkEditor

*CkEditor* es un editor de text HTML en línia, gratuït i de codi obert. Aquest està ideat per desenvolupadors i ofereix moltes possibilitats de personalització. Serà l'eina que donarà suport el nostre projecte i que utilitzarem com a base per a creació del nostre plugin. S'ha escollit *CkEditor* per la quantitat de documentació que ofereix tant a la seva pàgina oficial com a diversos fòrums d'Internet i per les possibilitats que ens dóna el seu alt nivell de personalització.

Per la realització d'aquest projecte s'utilitzarà la versió 4.6.2, que és la versió final que hi ha per descarregar a la seva pàgina oficial.

### Estat

L'objectiu final d'aquest projecte és aconseguir que els joves trobin una motivació extra en l'aprenentatge i augmentin així les seves ganes per seguir en el món escolar. Si això succeeix, pot representar en un futur un descens del fracàs escolar que comporta una societat més preparada pel món laboral.

## 3. Abast del projecte

Per tal que el projecte tingui el pes necessari per a un TFG s'ha acotat un nombre de funcionalitats que haurà de complir per estar assolit en la seva totalitat. Tenint en compte el temps del qual es disposa per la realització del treball i contemplant els possibles obstacles que apareixeran durant la realització del projecte, s'ha decidit que algunes funcionalitats siguin de caràcter opcional per assegurar-se que es compleix el que es menciona en el document.

El projecte consta de dues parts diferenciades:

- Un editor HTML online al qual s'afegirà el nostre *plugin*.
- Un entorn HTML+Javascript que executarà el codi generat pel *plugin*.

### 3.1 *Plugin* de mapeig d'imatges

El *plugin* afegit a l'editor ens introdueix la possibilitat de marcar zones d'una imatge prèviament inserida i associar a aquestes zones un element de pregunta-resposta. S'oferiran diverses eines per marcar les zones a la imatge:

- **Mitjançant un cercle.** L'usuari podrà generar el mapa dibuixant un cercle que podrà modificar al seu gust.
- **A mà alçada.** L'usuari podrà dibuixar amb el seu ratolí la forma que vol que tingui el mapa que vol inserir.
- **Amb una figura definida.** Es donarà a l'usuari un ventall de figures que podrà col·locar sobre la imatge. A l'inserir-la es generarà automàticament un mapa al voltant de la figura que serà modificable. Quan el mapa estigui definit, es donarà l'opció de fer la figura visible o no per l'usuari que hagi de realitzar l'activitat. Si aquesta figura no és visible serà l'usuari que resolgui l'activitat qui haurà d'encertar aquesta figura dins el radi d'acció prèviament definit.
- **Vareta màgica.** Es donarà la possibilitat a l'usuari de generar un mapa fent click sobre un punt de la imatge, i es seleccionarà una zona que comparteixi el mateix color que el píxel on l'usuari ha fet click amb el ratolí. Aquesta funcionalitat la comparteixen molts editors d'imatges amb la icona d'una vareta màgica. Donat que aquesta funcionalitat pot arribar a tenir una major complexitat i no sabem les limitacions que ens trobarem amb el CkEditor, aquesta funcionalitat quedarà dins les funcionalitats opcionals.

En el moment que un usuari generi un mapa sobre la imatge se li donarà l'opció de crear un quadre de text, al qual li introduirà una pregunta i una resposta i se li assignarà al mapa prèviament creat. El format de la resposta serà tipus test o escrit.

Quan l'usuari decideixi que ha finalitzat l'edició de l'activitat aquesta imatge es guardarà amb la informació dels mapes i el quadres de text assignats a ells. En aquell moment l'activitat estarà llesta perquè algú la intenti resoldre.

El que veurà l'alumne quan hi accedeixi serà una imatge amb un botó que li permetrà iniciar l'activitat.

## **3.2 Activitat interactiva**

La segona funcionalitat que oferirà aquest projecte és la possibilitat de resoldre d'una forma interactiva l'activitat creada amb el plugin explicat prèviament. L'usuari veurà per una banda tota la informació que hi hagués abans en l'editor, com pot ser un text explicatiu de l'activitat, i per l'altre, les preguntes sense la seva resposta

Quan estigui dins l'activitat l'usuari haurà de clicar sobre els mapes distribuïts per tota la imatge. En fer-ho, li apareixerà el quadre de text amb la pregunta escrita prèviament per qui ha realitzat l'exercici. Llavors quan l'usuari respongui rebrà un feedback informant-lo si la seva resposta és o no és correcta i la zona del mapa on es troba se senyalitzarà amb un color adient a l'estat de la pregunta.

Quan l'usuari contesti correctament totes les preguntes finalitzarà l'exercici i es mostrarà el nombre d'encerts i errors que ha tingut.

Una funcionalitat opcional que es planteja en aquest projecte és la possibilitat de donar l'opció de donar un feedback al realitzador de l'exercici de les qualificacions obtingues per un usuari utilitzant alguna eina d'emmagatzemat de dades en línia o enviant-li un correu.

## 3.3 Metodologies a utilitzar

### Mètode de treball

Donat que aquest treball només el realitzarà una persona seguint les indicacions que rebrà del director, i no s'ha de treballar en equip ni fer reunions amb cap client, s'optarà per seguir una metodologia de treball clàssica. El document d'anàlisi de requisits s'extraurà de tot el treball realitzat a l'assignatura de GEP, i es realitzarà el disseny i la codificació a l'hora d'entregar la memòria del projecte, també es realitzaran i documentaran els diferents tests que es realitzin durant el projecte.

Per tal d'acabar el projecte dins el termini amb totes les funcionalitats esmentades i donat que no s'utilitzarà una metodologia àgil basada en sprints. S'assignaran tres fites dins la planificació temporal on s'assignaran unes certes funcions dins la planificació temporal. Tot el que es faci en el projecte tindrà la seva documentació adient i es presentarà amb la memòria del projecte.

Es realitzaran reunions amb el director cada dues setmanes i es mantindrà una comunicació continuada via correu electrònic per tal d'assegurar-se que es segueix la planificació acordada i per poder encaminar el projecte en la bona direcció.

### Eines de seguiment

Per tal realitzar el projecte correctament s'utilitzarà l'eina de control de versions Git, que encara que el projecte sigui individual, és molt útil a l'hora de mantenir un codi ordenat, ben documentat i tenir un històric en cas de pèrdues de dades.

S'utilitzarà el correu electrònic com a eina de comunicació entre l'estudiant i el director i l'editor de *Google Drive* quan sigui necessari compartir algun document o fer un seguiment de la documentació del projecte.

S'habilitarà un servidor web on es penjarà l'última versió del projecte, perquè estudiant i director puguin fer proves i buscar errors.

També s'utilitzarà *Jira* per fer un seguiment de les funcionalitats que es vagin desenvolupant en el projecte i per informar dels possibles *bugs* que es vagin trobant durant el procés.

## **4. Planificació temporal**

### **4.1 Duració del projecte**

El Treball de fi de grau plantejat en aquest projecte té una data establerta per la seva realització. El projecte s'inicia a mitjans de febrer i es presenta davant el tribunal a finals de juny. S'espera que el projecte estigui acabat a finals de maig, però donat que poden succeir problemes que puguin afectar a la planificació temporal i que s'ha de preparar la memòria i la presentació final, s'emprarà el mes de juny per aquests últims retocs.

### **4.2 Divisió del projecte**

El projecte es dividirà en 6 parts diferenciades: una de planificació, una per preparar l'entorn del projecte, tres pel desenvolupament del projecte en si mateix i la fase final per acabar la documentació del projecte i per preparar la presentació.

En aquest apartat passarem a explicar les diferents fases i les tasques que tenen cadascuna. Primer es farà una breu explicació de la fase i quines funcionalitats es pretenen aconseguir en ella. I després s'extrauran les tasques pertinents a les funcionalitats que es volen crear i s'assignarà un nombre aproximat d'hores. El nombre d'hores assignat a cada tasca serà una potència de 2 depenen de la dificultat que es vegi per la seva realització (Exceptuant el curs GEP que té un nombre d'hores determinat).



## Planificació del projecte:

En aquest apartat tenim, per una banda tot el que es realitza dins el curs de GEP, abast i context del projecte, planificació temporal, recursos que s'utilitzaran, estimació de costos, sostenibilitat i les dues presentacions que es realitzaran.

Per l'altra banda, fora del que es realitza en el curs de GEP, tenim l'especificació de les funcionalitats del projecte i el disseny de les interfícies que hi haurà en ell i com es relacionaran amb les funcionalitats d'aquest.

Tasca	Hores
Planificació dins del curs GEP	80
Especificació de les funcionalitats del projecte	16
Disseny de les funcionalitats del projecte	16
Disseny de les interfícies gràfiques del projecte	16
TOTAL	128

## Fase Inicial:

En aquesta fita ens centrarem a posar en marxa el projecte. És a dir, preparar l'entorn en el qual es treballarà durant tot el projecte. Per això caldrà muntar un servidor web que utilitzarem per realitzar les proves pertinents al projecte. Per poder realitzar les proves esmentades es caldrà que hi hagi una pàgina web senzilla on s'inclourà l'editor de text CkEditor.

Quan l'entorn estigui preparat serà necessari un temps per poder fer proves amb els plugins de CkEditor i intentar crear-ne alguns senzills per tenir els conceptes clars abans d'iniciar el projecte.

Tasca	Hores
Muntar un servidor	4
Crear una petita pàgina web	8
Incloure <i>CkEditor</i> dins la pàgina	4
Testejar funcionament dels <i>plugins</i> dins <i>CkEditor</i>	16
Crear el <i>plugin</i> i incloure'l dins l'editor	4
Documentar	4
TOTAL	40

### Primera fase de desenvolupament:

Quan l'entorn de proves ja estigui preparat, serà el moment per començar a desenvolupar el projecte. S'ha decidit dividir el període de desenvolupament en tres fases per diverses raons. Si existeix una bona planificació de les tasques que s'han de realitzar en un cert temps facilita al desenvolupador a organitzar-se millor per tal de tenir les tasques en el temps proposat. Una bona raó per la qual és útil dividir-ho en fases és que el director pugui tenir un millor seguiment del projecte i que l'estudiant tingui més objectius a curt pla.

En aquesta primera fase es vol aconseguir un primer prototipus del projecte. Caldrà que el projecte compleixi les seves funcionalitats més bàsiques. Que l'usuari pugui marcar una zona de la imatge, d'una manera senzilla, com pot ser amb una eina quadrada. I que es pugui assignar un quadre de text, amb la seva pregunta i resposta dins la zona.

També en aquesta fase es desenvoluparà el que veurà l'estudiant, que serà el botó d'execució de l'activitat, que generarà una nova pestanya amb l'activitat interactiva. En aquesta primera fase es generarà el quadre de text quan l'estudiant faci click a la zona marcada pel professor i que pugui donar una resposta, que es marcarà com a correcta o errònia.

Tasca	Hores
Crear una eina per marcar una zona en la imatge	8
Crear un panell de configuració de la zona	4
Generar un quadre de text amb una pregunta i una resposta	8
Assignar el quadre de text a la zona creada	4
Guardar la informació de la zona prèviament creada dins l'HML de l'editor	16
Assignar un botó d'execució que generarà l'activitat en una nova pestanya	32
Generar el quadre de text quan es faci <i>click</i> sobre la zona marcada	8
Generar feedback quan encertes/falles una resposta	2
Testejar i documentar conjunt de funcionalitats	8
TOTAL	90

### Segona fase de desenvolupament:

En aquesta segona fase ens ocuparem d'augmentar la personalització de l'activitat, donarem més opcions a l'usuari per generar zones a la imatge que s'adaptin a l'activitat que volen oferir.

Tasca	Hores
Donar diverses opcions per generar mapes a l'usuari (cercle, mà alzada, pinzell)	32
Opció assignar un quadre a múltiples mapes	4
Ampliar panell de configuració dels mapes	2
Donar opció de ressaltar la zona <i>mapejada</i>	8

Ampliar panell de configuració quadres de text	2
Afegir opció d'inserir múltiples respostes vàlides	4
Donar opció de generar pregunta tipus test	8
Testejar i documentar conjunt de funcionalitats	8
TOTAL	68

### Tercera fase de desenvolupament:

En aquesta fase donat que ja es tindrà més experiència a l'hora de programar en *JavaScript* i en dissenyar *plugins* per *CkEditor* es passarà a realitzar les tasques més complexes, i també les tasques opcionals.

Tasca	Hores
Generar mapes utilitzant figures amb un radi d'acció	16
[opc.] Generar mapes per colors iguals ( <i>vareta màgica</i> )	16
Desenvolupar l'activitat utilitzant figures	16
Donar opció personalitzar feedback	8
Opcions avançades	16
[opc.] Guardar informació del resultat d'un estudiant a una base de dades externa	16
Mostrar resultat de l'activitat	4
Fer un disseny atractiu per les icones i les imatges del projecte	8
Testejar i documentar conjunt de funcionalitats	8
Realitzar optimitzacions pertinents del projecte	16
TOTAL	124

### Fase final:

En aquesta fase s'acabarà de documentar la memòria del projecte i es prepararà la presentació.

Tasca	Hores
Acabar memòria	32
Preparar presentació	32
TOTAL	64

### Hores totals:

Fase	Hores
Planificació del projecte	128
Fase Inicial	40
Primera fase de desenvolupament	90
Segona fase de desenvolupament	64
Tercera fase de desenvolupament	124
Fase final	64
TOTAL	510

## 4.3 Valoració d'alternatives i pla d'acció

### Riscos

Durant el desenvolupament del projecte poden succeir molts problemes diferents. Per una banda es podria arribar a tenir problemes amb la falta de coneixement del llenguatge, i per l'altre, el fet d'estar treballant sobre *CkEditor* i les seves limitacions, pot arribar a donar problemes que ens costi molt temps de solucionar.

### Pla d'acció

Per la correcta realització del projecte s'ha calculat que hi ha temps suficient per realitzar totes les tasques abans de la seva data de finalització. Però donat que coneixem els riscos que hi pot haver s'ha de treballar tenint-los en compte.

Si alguna de les tasques proposades ocupés més temps del desitjat, no seria un problema, ja que s'han definit tasques opcionals, i es podria aprofitar el temps d'aquestes en realitzar correctament alguna tasca més costosa.

Si per un altra banda hi hagués alguna tasca que no es pogués arribar a realitzar per alguna raó, ja fos perquè les limitacions del *CkEditor* no ho permeten o per falta de coneixements en la matèria és podria replantejar sempre que no afectés una funcionalitat bàsica.

## 4.4 Recursos

Per la realització del projecte seran necessaris una sèrie de recursos.

- **Recursos humans**, que seran un estudiant encarregat de desenvolupar projectar i realitzar una documentació adient, i un professor que s'encarregarà de dirigir a l'estudiant quan ho necessiti i fe un seguiment del projecte.
- **Recursos materials**, que només s'utilitzarà un portàtil amb Windows 10 i connexió a internet.
- **Recursos software**, s'utilitzarà *IntelliJ* com a entorn de programació, es muntarà la pàgina web sobre un servidor *Apache* muntat amb l'eina *Wamp*, i es farà servir l'eina Word del paquet office per realitzar la documentació del projecte.

## 5. Gestió econòmica

### 5.1 Identificació dels costos

Ara que ja sabem quin son els recursos que necessitarem per desenvolupar el projecte, haurem de realitzar la seva planificació econòmica. Dividirem el cost de la mateixa manera, si vénen de recursos humans materials o Software.

### 5.2 Costos humans

Com ja s'ha realitzat una prèvia planificació temporal del projecte i s'ha extret el nombre d'hores que s'invertiran en cada tasca, l'únic que resta per fer és traduir aquesta inversió temporal en un cost econòmic.

Per intentar que el projecte que es presenta en aquest document s'assembli més a un projecte empresarial, suposarem que les tasques que es realitzen en ell seran realitzades per tres persones amb tres càrrecs diferents. Un cap de projecte, encarregat de la planificació, la documentació i la preparació de la presentació als possibles compradors, un programador *Senior* encarregat de desenvolupar les tasques més complexes, i un programador *Junior* encarregat de testejar les funcionalitats i desenvolupar les tasques més senzilles.

Càrrec	Hores	€/Hora	Total
Cap de projecte	192 h	14 €/h	2688 €
Programador <i>Senior</i>	178 h	12 €/h	2136 €
Programador <i>Junior</i>	140 h	8 €/h	1120 €
			5.944 €

### 5.3 Costos materials i Software

Per les tasques que volem realitzar necessitarem un portàtil 15,6" ASUS K501UQ-DM012T (només es tindrà en compte un cost proporcional amb l'ús que es donarà al projecte), connexió a Internet i un servidor per fer proves durant 3 mesos, el IDE *IntelliJ* i l'eina *Word* del paquet Office.

Material	Preu (€)
Portàtil 15,6" ASUS K501UQ [0,25]	175 €
Internet (3 mesos)	60 €
IntelliJ (1 any)	499 €
Word 2016	103,20 €
	912,19 €

### 5.4 Possibles imprevistos

Com sabem que poden haver-hi imprevistos no desitjats en el projecte cal tenir en compte que el pes econòmic que poden tenir sobre el nostre projecte.

Durant el desenvolupament del projecte es podria donar el cas que alguna tasca durés més temps del previst a la planificació, per arribar al termini acordat significarà que s'hauran de realitzar hores extres per dur a terme la tasca en qüestió. Plantejarem que aquesta desviació temporal serà com a màxim d'un 10% de les hores invertides al desenvolupament. Això es representa amb 32 h ( $318 \text{ h} + 318 \text{ h} * 0,1$ ) que s'atribuiran al desenvolupador Senior i suposaran un augment de 384 € ( $32 \text{ h} * 12 \text{ €}$ ).

Un altre possible imprevist seria que s'espatllés l'ordinador durant el procés de desenvolupament o documentació. Com que tot el projecte es trobarà al núvol, sigui a Github per el codi o a Drive per la documentació, no suposarà greus pèrdues temporals (sempre hi haurà una petita part en local que sí que es podria arribar a perdre, però suposarem que és negligible), però sí que implicaria que s'hauria d'invertir una certa quantitat en un altre ordinador.

Per finalitzar, per assegurar-nos que no ens quedem curts amb la planificació econòmica s'aplicarà un percentatge de contingències al cost total d'un 5%.

## 5.5 Costos totals

Costos	Preu (€)
Costos humans	5.944 €
Costos materials	912,19 €
Imprevistos	901,81 €
> Imprevistos humans	384 €
> Imprevistos materials	175 €
> Contingències	342,81 €
	7.758 €

## 5.6 Retorn de la inversió

L'objectiu d'aquest projecte és crear una eina que doni als docents una oportunitat per augmentar la motivació que tenen els seus alumnes sense haver de fer cap inversió prèvia. És per aquesta raó que el producte que es proposa en aquest projecte és de lliure utilització per tothom que ho necessiti a cost zero.

Però tenint en compte que existeix una inversió en aquest projecte i s'ha de recuperar es proposa orientar-la en subvencions per part de l'estat i les entitats educatives per una part i subvencions personals per l'altre.

Per una banda quan el producte estigui en circulació s'oferirà una pàgina de contacte en la informació del plugin on es donarà la possibilitat als usuaris de donar una quantitat econòmica per tal de recuperar la inversió realitzada en el projecte. Per l'altre es parlarà amb entitats educatives i governamentals per tal que donin suport econòmic al projecte.



## 6. Solució proposada

Per resoldre els problemes plantejats en la primera part del document i assolir els objectius enumerats durant aquest projecte es desenvoluparà una eina d'edició d'activitats interactives, que s'anomenarà QuizEditor i que estarà incorporada com a *plugin* dins l'editor de text en línia *CkEditor*.

Aquest constarà de dos botons (Fig 1) que representaran les dues parts en les quals és divideix el projecte. El primer botó serà l'editor d'activitats interactives, que obrirà una finestra on es podrà editar l'activitat sobre una imatge. El segon botó serà el que executarà l'activitat prèviament creada.



Icona d'edició d'imatges



Icona d'execució d'activitats

Fig. 6.1 Icones utilitzades pels botons del plugin

Quan es faci clic sobre el primer botó, si hi ha una imatge seleccionada, s'obrirà una pantalla on s'iniciarà l'edició de l'activitat. Si no és així, donarà un missatge d'error informant que falta aquesta imatge. En aquesta pantalla apareixerà per una banda la imatge on es vol introduir l'activitat interactiva i les eines disponibles per realitzar l'edició. (Fig 2).



Fig. 6.2 Pantalla inicial de l'eina d'edició

En seleccionar una eina es podrà començar a pintar sobre la imatge la zona on es vol inserir una activitat, quan aquesta zona estigui finalitzada s'habilitaran dos quadres de text. El primer serà la pregunta que apareixerà quan l'estudiant seleccioni aquella zona, i el segon serà la resposta que l'estudiant haurà d'introduir perquè la pregunta sigui correcta (Fig. 3).

En aquest moment es permetrà configurar dos paràmetres. El tipus de pregunta, si serà una pregunta de resposta única o de tipus test (Fig. 4). I es donarà també l'opció de fixar la pregunta, per si l'usuari vol que sigui igual en totes les preguntes i no haver d'escriure constantment el mateix (Fig. 3).



Fig. 6.3 Paràmetres de configuració de la pregunta

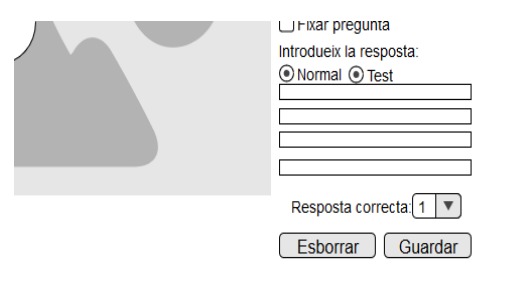


Fig.6. 4 Configuració de preguntes test

En aquest moment l'usuari podrà guardar la pregunta per assignar-la a la zona que ha marcat prèviament o marcar una altra zona on s'assignarà la mateixa pregunta. Aquesta opció es podrà repetir tantes vegades com vulgui l'usuari fins que decideixi guardar les zones marcades (Fig. 5).



Fig. 6.5 Botó per marcar més zones de la imatge

Quan el docent cregui que la activitat està finalitzada haurà de finalitzar l'edició.

Per acabar l'activitat el docent tindrà a la seva disposició totes les eines que li ofereix l'editor *CkEditor* per poder introduir un text introductor a l'activitat, donar-li un títol o inserir el temari necessari per poder realitzar l'activitat a consciència.

En el moment en que l'activitat estigui finalitzada es podrà executar per part de l'estudiant a qui vagi dirigida. Dins la pantalla d'edició haurà de seleccionar la imatge que prèviament ha sigut editada. Al fer clic en el segon botó que dóna a disposició aquest *plugin* obrirà una nova pestanya en el navegador on apareixerà un botó que permetrà a l'alumne iniciar l'activitat.

Quan l'estudiant premi el botó apareixerà la imatge amb les zones marcades anteriorment per el docent. Sortirà també el nombre de preguntes que té la imatge en qüestió i quantes preguntes s'han encertat durant la realització de l'exercici, que en aquesta fase inicial seran 0.

Quan l'usuari que realitza el rol d'estudiant seleccioni una de les àrees marcades a la imatge li apareixerà a la pantalla un quadre de text amb una pregunta, anteriorment inserida per el professor que ha realitzat l'activitat, i una zona de text on podrà afegir una resposta (Fig. 6).

## 0/2 Respostes correctes

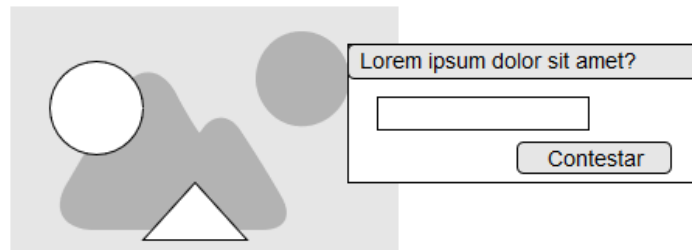


Fig. 6.6 Quadre de text al clicar una zona marcada

Si la resposta afegida es correcta rebrà un *feedback* per part de l'aplicació, indicant-li si aquesta resposta es o no correcta. Aquest feedback es veurà amb un codi de colors, verd si encerta, vermell si falla, que apareixerà tant a la part superior de la pantalla com en l'àrea de la imatge que acaba de respondre. Quan falli una de les preguntes apareixerà un comptador a la pantalla indicant el nombre d'errors que ha produït l'estudiant durant la realització de l'exercici. (Fig. 7).

**¡Resposta incorrecta!**

0/2 Respostes correctes

Errors : 1

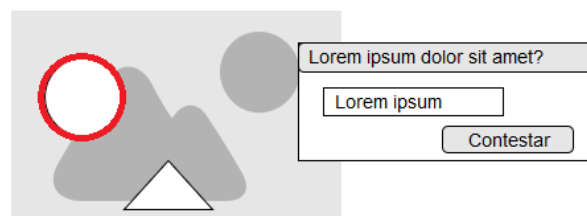


Fig. 6.7 Feedback al respondre una pregunta incorrectament

L'estudiant tindrà l'oportunitat de repetir una pregunta fins que sigui capaç d'encertar-la, en aquell moment desapareixerà el requadre i se li mostrarà un *feedback* positiu indicant-li que la resposta es correcta. (Fig. 8).

**¡Resposta correcta!**

1/2 Respostes correctes

Errors : 1

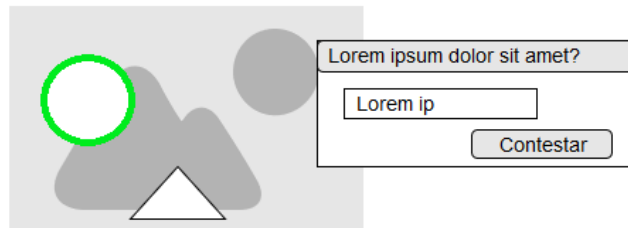


Fig. 6.8 Feedback al respondre una pregunta correctament

## 7. Per què *CkEditor*?

Per la realització d'aquest projecte s'ha decidit integrar l'eina proposada dins un editor de text en línia, en el nostre cas, s'ha optat per utilitzar CkEditor. Aquest editor ofereix una gran quantitat de llibreries als desenvolupadors, una bona documentació a la seva pàgina web i una àmplia comunitat a les seves espatlles.

La decisió d'utilitzar un editor de text en línia com a servei per oferir l'eina que es treballa en aquest projecte té una sèrie d'avantatges i d'inconvenients, tant pels usuaris que utilitzaran aquesta eina com pel desenvolupament d'aquest projecte. Donat que l'editor és un factor imprescindible en aquest projecte serà essencial parlar de totes les utilitats que ha aportat.

### 7.1 Avantatges

Per començar a parlar dels avantatges d'utilitzar un editor de text en línia per la realització d'aquest projecte cal entendre a quins usuaris va dirigit. Hi haurà dos tipus d'usuaris que veuran utilitat en utilitzar aquesta eina. Per una banda estaran els docents que poden, o no, tenir una pàgina web on incorporar un editor de text en línia i configurar en ella diverses activitats interactives utilitzant el *plugin* desenvolupat en aquest projecte. Com acabem de comentar en l'apartat anterior pot ser que un docent no disposi dels recursos necessaris per utilitzar *QuizEditor*, i és aquí on entra el segon tipus d'usuaris, desenvolupadors de pàgines web orientades en oferir editors d'activitats interactives per professors. Aquests seran els que inclouran aquest *plugin* dins la seva eina d'edició i conjuntament amb altres eines de les quals disposin oferiran als docents la possibilitat de realitzar activitats i enviar-les a l'alumnat mitjançant el seu portal web.

El fet de tenir l'eina integrada en un editor aporta a l'usuari la facilitat d'accedir en qualsevol moment al creador d'activitats. Només necessita entrar a la pàgina web on estigui integrat l'editor, per una banda, i el *plugin* instal·lat, per l'altre, i ja pot començar a preparar les activitats que proposarà posteriorment als seus alumnes.

Per els usuaris que vulguin incorporar l'editor dins el seu propi portal web començar trobaran que és una tasca senzilla i es pot trobar ràpidament informació adient per integrar-ho en qualsevol sistema. Actualment és molt comú, sobretot en webs on interactuen els estudiants, que es disposi d'un editor de text d'aquesta mena. Per exemple, el campus virtual de la FIB, el RACÓ, utilitza l'editor *tinyMCE* per realitzar moltes de les seves gestions.

Utilitzant editors en línia d'aquesta mena, comunament anomenats WYSIWYG (*What You See Is What You Get*), dóna l'oportunitat a l'usuari de crear un text en format HTML sense la necessitat de conèixer el llenguatge, i en molts casos sense tenir la noció que l'estàs utilitzant. Et permet realitzar gran quantitat d'accions com afegir capçaleres

(*headers*), remarcar paraules o inserir imatges dins del text que estàs escrivint, que automàticament es podrà guardar en format HTML per visualitzar-se posteriorment en el mateix format. Aquestes funcionalitats donen un gran suport a un eina com aquesta ja que permeten la possibilitat d'incorporar una activitat conjuntament amb un text que aporti uns conceptes útils per realitzar-la.

Cal destacar que gràcies a estar integrat dins un editor en línia amb una comunitat tan gran com la que disposa *CkEditor* et dóna la possibilitat d'utilitzar una immensa quantitat de *plugins* que poden ser d'utilitat per complementar la utilitat de l'eina en qüestió.

## 7.2 Inconvenients

També és important destacar que utilitzar un editor de text com el que es proposa en aquest projecte aporta també alguns inconvenients. Per contraposició amb els avantatges comentats anteriorment, el fet que aquesta eina necessiti d'un editor de text en línia i no sigui una senzilla aplicació d'escriptori fa que sigui imperativa la necessitat de disposar d'una pàgina web per a l'usuari que la vagi a utilitzar. I això, pot allunyar a algun usuari que necessiti les funcionalitats que ofereix i no tingui ningun recurs per utilitzar-lo.

El fet d'utilitzar *CkEditor*, per molt que aporti una gran quantitat de possibilitats de vegades limita les funcionalitats que et pot oferir un simple llenguatge de programació de *JavaScript* amb HTML i els desenvolupadors han de buscar, en molts casos, alternatives a l'extensa quantitat de funcions que aporta la seva API.

## 7.3 Conclusions

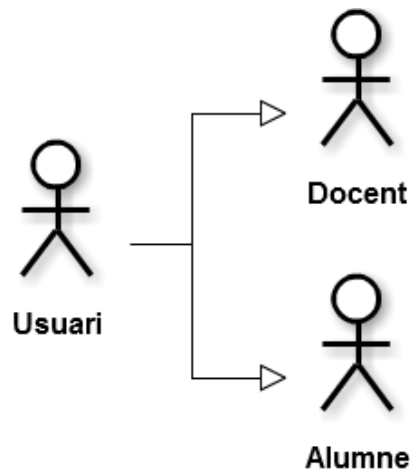
Realitzar una aplicació com la que es presenta en aquest document sobre una plataforma com és *CkEditor* i tenint en compte les avantatges i els inconvenients estudiats anteriorment en aquest apartat per un projecte d'aquesta categoria i els usuaris a qui va dirigit, es un factor positiu per aquest. Ja que ens ofereix un eina fàcilment integrable en qualsevol portal web i aporta moltes possibilitats de tenir en el mateix editor diversos *plugins* amb la mateixa finalitat i oferir dins *CkEditor* una eina molt potent per crear activitats interactives de tota mena i adaptables a qualsevol àmbit d'estudi de l'alumne.

# 8. Planificació del projecte

## 8.1 Casos d'us

Per començar a parlar del disseny del projecte i abans de realitzar el desenvolupament d'aquest és bo pensar en les funcionalitats que necessitaran els diferents usuaris a l'hora d'utilitzar l'aplicació que es presenta en aquest projecte. Una forma clara de veure aquestes diferents funcionalitats es presentant els casos d'ús que tindran els usuaris a l'hora d'interactuar amb el projecte.

Per començar es dividiran els usuaris en dos grups, tenint ten compte el rol que tindran en utilitzar l'aplicació: els **docents**, que seran els encarregats d'editar l'activitat que posteriorment serà executada, i els **alumnes**, que seran els que executaran l'activitat preparada pels docents.



## Casos d'ús de docent

Els casos d'ús assignats al usuari amb rol de docent son els següents:



**Crear una activitat:** L'usuari podrà iniciar l'edició d'una activitat sobre una imatge prèviament inserida a l'editor de text.

**Precondició:** L'usuari ha de tenir una imatge seleccionada per començar l'edició.

**Editar una activitat prèviament creada:** L'usuari podrà editar una activitat que prèviament hagi creat ell o un altre usuari.

**Precondició:** L'usuari ha de tenir una imatge seleccionada amb una activitat creada per QuizEditor per començar l'edició.



**Escollir un eina de dibuix:** Durant el procés d'edició d'una activitat, l'usuari podrà escollir un eina de dibuix per marcar una zona dins de la imatge.

**Precondició:** El sistema no està esperant a que l'usuari acabi de marcar una zona.

**Marcar una zona de la imatge:** L'usuari podrà marcar qualsevol zona de la imatge amb una de les eines que aporta l'editor.

**Precondició:** L'usuari ha escollit prèviament una eina de dibuix.

**Veure la informació d'una zona marcada:** Quan l'usuari seleccioni una zona de la imatge el sistema li mostrarà la informació (pregunta/resposta) que conté aquella zona.

**Eliminar una zona de la imatge:** L'usuari podrà eliminar una zona prèviament marcada de la imatge.

**Precondició:** Ha d'existir una zona seleccionada a la imatge.

**Afegir una pregunta a una zona marcada:** L'usuari podrà escriure una pregunta que apareixerà quan l'estudiant faci clic sobre la zona seleccionada.

**Precondició:** Ha d'existir una zona seleccionada a la imatge.

**Afegir una resposta a una zona marcada:** L'usuari podrà escriure una resposta sobre una zona seleccionada que serà la resposta que l'estudiant haurà de contestar quan li aparegui la pregunta al seleccionar aquella zona.

**Precondició:** Ha d'existir una zona seleccionada a la imatge.

**Assignar una pregunta/resposta a una zona:** L'usuari podrà guardar la informació que contindrà una determinada zona de la imatge.

**Editar una pregunta/resposta:** Quan l'usuari estigui visualitzant la informació d'una zona de la imatge tindrà la opció d'editar la informació d'una zona, ja sigui modificant la pregunta, la resposta, el tipus de resposta, o fins i tot d'editar la zona prèviament marcada en aquella imatge.

**Precondició:** Ha d'existir una zona seleccionada a la imatge.

**Marcar més d'una zona per a una pregunta mateixa pregunta:** Quan l'usuari acabi de dibuixar una zona de la imatge tindrà la opció de marcar-ne una altre i tornar al cas d'us d'escollir un eina de dibuix.

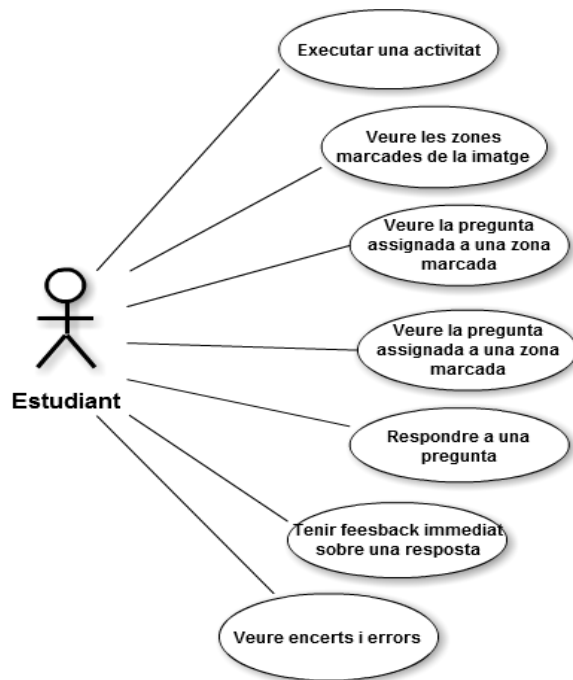
**Escollir pregunta tipus test:** L'usuari podrà escollir si la pregunta que vol assignar a una zona serà de resposta única o tipus test. Per defecte sempre serà resposta única.

**Finalitzar edició:** Quan l'usuari cregui que l'activitat està acabada podrà finalitzar l'edició d'aquesta i els canvis es guardaran. En aquest moment la imatge estarà llesta per executar-se.

**Veure les zones marcades de la imatge:** Durant la pantalla d'edició el sistema mostrarà tots els canvis que es vagin produint durant l'edició de la imatge i es mostraran totes les zones que aquesta te marcades.

## Casos d'us d'estudiant

Els casos d'us assignats a un usuari amb rol d'estudiant son els següents:



**Executar una activitat:** Un estudiant podrà executar una imatge que estigui dins l'editor i que prèviament hagi estat modificada amb QuizEditor.

**Precondició:** Que existeixi una imatge seleccionada amb un usemap assignat per QuizEditor.

**Veure les zones marcades de la imatge:** Quan l'estudiant executi l'activitat veurà una pantalla on apareixeran totes les zones prèviament marcades pel docent.

**Veure la pregunta assignada a una zona marcada:** L'usuari veurà un requadre amb la pregunta corresponent a la zona de la imatge sobre la qual hagi fet clic.

**Respondre a una pregunta:** L'usuari podrà respondre a una pregunta que li aparegui en la pantalla, associada a una zona de la imatge.

**Tenir feedback immediat de la resposta:** En el moment que un usuari contesti a una pregunta li apareixerà en la pantalla si aquesta ha estat correcta o incorrectament contestada, també la zona sobre la qual hagi contestat li apareixerà en un color o un altre tenint en compte si la resposta era o no correcta.

**Veure encerts i errors:** A la pantalla que se li mostrarà a l'usuari després d'executar l'activitat podrà veure els encerts que ha tingut sobre les preguntes totals i els errors que ha comès durant la resolució de l'exercici.

## 8.2 Requisits no funcionals

### Facilitat d'us (Usabilitat):

L'aplicació ha de ser fàcil d'usar i senzilla per a qualsevol usuari sense necessitat de tenir coneixements previs. No serà necessari haver de recordar ningun procediment preestablert per poder utilitzar l'aplicació.

És important complir aquesta característica perquè l'eina que es proposa va dedicada a qualsevols docent en qualsevol àmbit estudiantil i pot ser que moltes vegades no estiguin acostumats a utilitzar eines d'aquesta mena.

### Intuïtiu i entenedor:

El sistema oferirà a l'usuari una línia de text informativa a la pantalla indicant-li algunes pautes per utilitzar l'eina que té al davant. Però aquesta funcionalitat, donat que es un ajut per l'usuari per realitzar amb més facilitat les tasques que permet l'aplicació, serà prescindible ja que l'aplicació serà suficientment intuïtiva i les seves funcionalitats seran ràpidament visibles per l'usuari.

### Adaptable:

L'aplicació serà adaptable a qualsevol sistema, donat que les dependències de les que disposa l'aplicació seran incloses dins aquesta. Així l'usuari que necessiti utilitzar-la no necessitarà integrar-les dins el seu portal web o descarregar-les d'un servei extern.

### Feedback continu:

L'aplicació oferirà constantment un feedback a l'usuari per les accions que estigui realitzant. Ja sigui a un docent, que rebrà informació a l'utilitzar les diferents funcionalitats que ofereix QuizEditor o a l'alumne que sempre veurà de manera entenedora si està, o no, responent una pregunta correctament.

## 8.3 Disseny previ

En la realització del disseny d'aquest projecte s'hi han realitzat dos diagrames previs al desenvolupament d'aquest. Primer s'han realitzat dos diagrames d'estat, un per cada usuari, els quals no han sofert pràcticament cap modificació durant el desenvolupament del projecte.

El segon diagrama realitzat ha estat el diagrama UML amb el disseny lògic del projecte, que aquest s'ha anat modificant durant el transcurs del projecte, a causa de la falta de coneixement del llenguatge utilitzat i de les funcionalitats de CkEditor.

## 8.4 Disseny UML de la lògica del problema

Per entendre el funcionament que tindrà l'aplicació és necessari realitzar un disseny de la seva arquitectura de classes. Aquest disseny ha existit des de l'inici del desenvolupament del projecte, i ha anat sofrint certes modificacions durant el seu creixement per poder adaptar-se amb més eficiència als requisits funcionals que ens demanava.

A banda del codi de configuració del *plugin* i de la plantilla HTML on és realitza l'activitat, s'han utilitzat quatre classes que contenen la lògica del projecte:

- **manager**: Classe encarregada de rebre les peticions de l'usuari i realitzar les operacions necessàries per resoldre aquestes peticions. El *manager* serà la classe que carregarà les pantalles d'edició i execució de l'activitat, inicialitzarà tots els elements, pintarà les zones seleccionades per l'usuari i emmagatzemarà la informació necessària pel desenvolupament de l'aplicació.
- **component**: Aquesta classe representa les àrees marcades per l'usuari, dins d'ella es guardarà la informació de la forma de la zona (*circle*, *shape* o *poly*) i les coordenades que ocupa en la imatge. La classe *component* representa els subelements HTML *area* que utilitza l'element *map*.

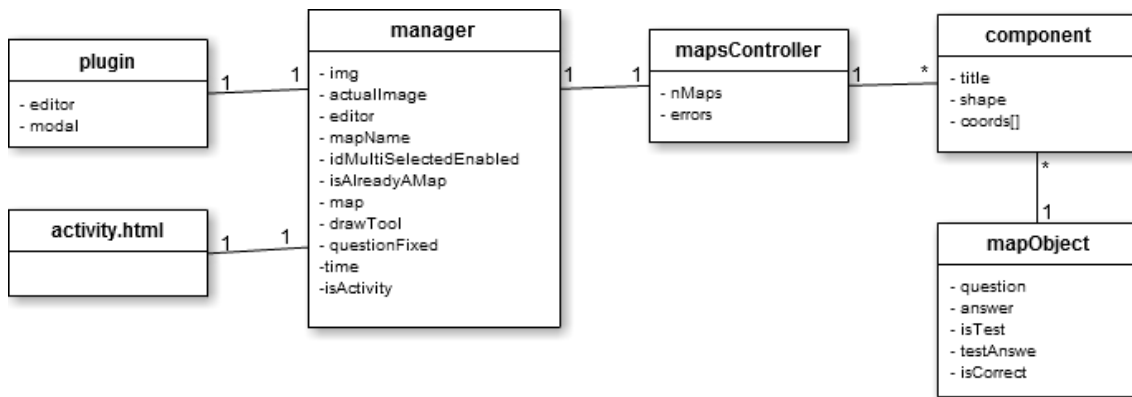
Per escollir el disseny correcte del projecte l'element *component* ha estat fonamental perquè és la classe amb major demanda del sistema, cada vegada que l'usuari fa clic sobre la imatge es comproven les diferents coordenades dels components per saber si ha seleccionat o no algun d'ells. A més, cada vegada que algun *component* és modificat, el *manager* tindrà que recórrer tota la llista per dibuixar-los sobre la imatge. És per aquestes raons que s'ha tingut molt en compte oferir un fàcil accés a la llista de *components* en la realització del disseny de la lògica de l'aplicació.

- **mapObject**: La classe *mapObject* representa la informació que contindrà una zona de la imatge. Durant l'edició de l'activitat emmagatzemarà la informació que l'usuari anirà introduint en la pantalla d'edició. En el moment d'executar

l'activitat enviarà al *manager* la informació d'aquestes i s'encarregarà de realitzar la lògica de resolució de les preguntes quan sigui necessari.

- **mapController:** Aquesta classe contindrà tots els elements *mapObject* i s'encarregarà d'aïllar la lògica relacionada per oferir els diferents elements de la imatge i emmagatzemarà informació rellevant per l'execució de les activitats.

## Primera proposta de disseny UML

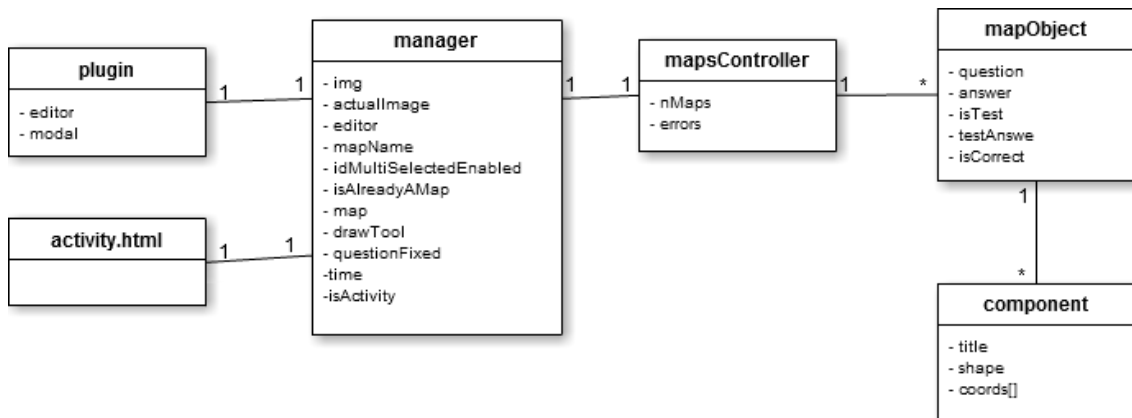


Aquest disseny va ser el primer que es va generar pel desenvolupament del projecte. En un principi era el disseny que millor enfocava la lògica del problema. Però no per aquesta raó era més eficient. El *controller* contenia tots els components i cadascun d'ells contenia la seva informació dins de la classe *mapObject*.

La classe *controller* tenia accés directe als diferents components. Això donava solució a l'alta demanda d'aquests elements.

Ràpidament es va observar el problema que aquest disseny suposava. Tenir una instància de *mapObject* per cada component del sistema suposava repetir informació en el moment de gestió i aportava dificultats a l'hora de mantenir la consistència de les dades. En aquest moment es va descartar la proposta de disseny.

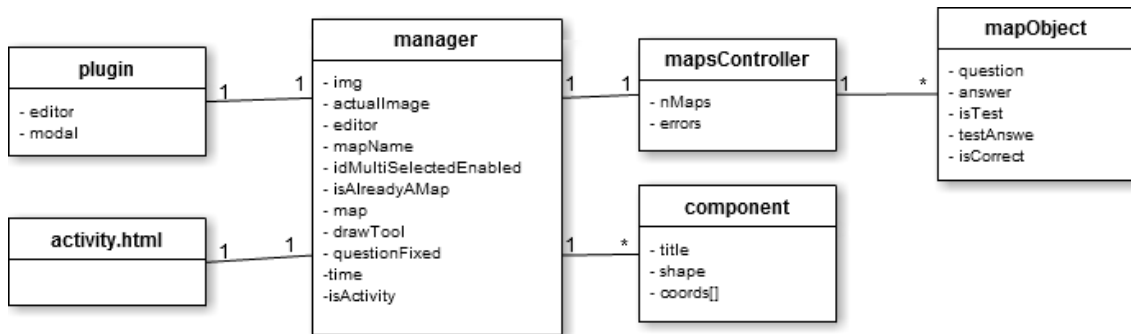
## Segona proposta de disseny UML



Aquest disseny solucionava el problema interior, el *controller* contenia una llista amb tots els elements *mapObject* i cadascun d'ells contenia varies instàncies de component. L'accés als diferents objectes de component era lleugerament més costós, però no suposava una alta carrega en el sistema.

A l'hora de desenvolupar aquesta solució es va veure el problema que aquest suposava. Les instàncies de *mapObject* es generaven posteriorment a la creació del component. Una solució podia ser crear un element auxiliar que contingués els diferents components que encara no tenien un *mapObject* assignat, però això suposava una petita inconsistència de les dades durant un període de temps. Per aquesta raó es va intentar buscar una solució alternativa.

## Disseny UML definitiu



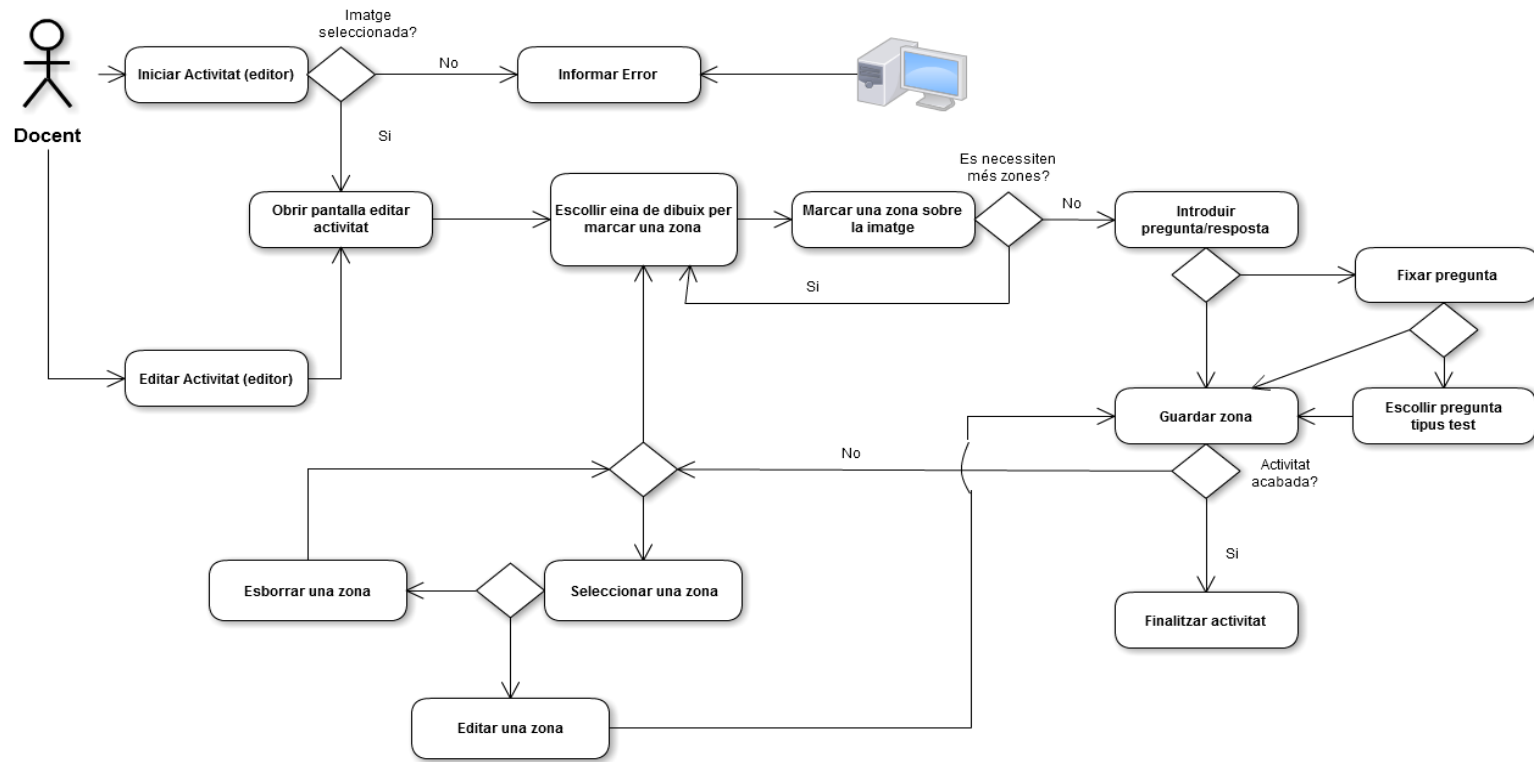
Aquest disseny solucionava els problemes trobats anteriorment. En aquest cas el manager coneix tots els components de la imatge i té la possibilitat d'accedir ràpidament a tots ells per afavorir l'alta demanda que aquesta classe suporta. I per l'altre costat conté tots els `mapObject` dins el controller.

Es va veure durant la realització del projecte que la necessitat d'accedir als diferents `mapObjects` era notablement inferior en comparació a la quantitat d'accessos de les dades dels components. Per aquesta raó en aquest disseny s'inclou una variable `id` als objectes `mapObjects` que serà referenciada a cada component al qual estigui associat. Amb aquest disseny es resolien els problemes dels anteriors afavorint al rendiment de l'aplicació.

## 8.5 Diagrama d'estats

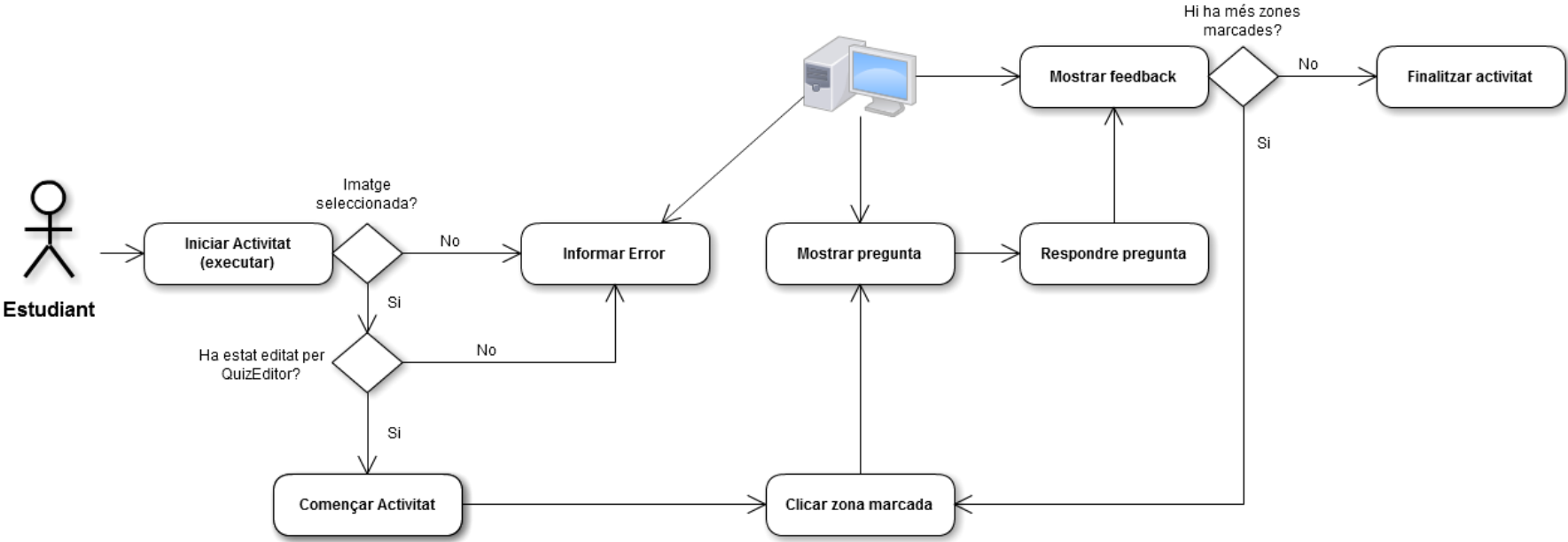
Una forma clara d'entendre el funcionament del programa es veient les diferents fases per les quals un usuari haurà de passar per realitzar les funcionalitats que permet realitzar l'aplicació. Aquest diagrama ha tingut molta rellevància durant el desenvolupament de l'aplicació. Ha estat, juntament amb el diagrama UML un factor a seguir al realitzar una nova funcionalitat del projecte.

En la pàgina següent es mostrarà el diagrama on apareixen les diferents fases per les quals passarà un usuari que realitzi el rol de docent en utilitzar QuizEditor.





En el següent diagrama es mostraran les diferents fases per les quals passarà un usuari que realitzi el rol d'estudiant a l'utilitzar QuizEditor:



## 9. Implementació

En aquest apartat es veurà com s'han desenvolupat les diferents seccions de codi del projecte, analitzarem el codi utilitzat per resoldre les seves funcionalitats i les comandes que necessàries utilitzades per integrar el *plugin* dins de *CkEditor*.

### 9.1 Paràmetres de configuració de *CkEditor*

Abans de desenvolupar el *plugin* que es vol presentar en aquest projecte ha sigut necessari crear una pàgina HTML senzilla on integrar *CkEditor*.

La idea proposada inicialment en aquest projecte era crear una pàgina web hostejada en un servidor per poder realitzar les proves de l'aplicació. Però finalment, s'ha decidit realitzar aquesta pàgina web però hostejada localment. D'aquesta manera ens estalviem el fet de configurar un servidor que a la pràctica no aportaria cap millora al projecte.

La pàgina que s'ha realitzat ha estat un senzill fitxer *HTML* amb només un títol amb el nom del projecte a la part superior i amb l'editor ocupant la resta de la pantalla. Amb això ens assegurem que no hi hagi informació innecessària que pogués destorbar a l'hora de realitzar proves.

Com s'ha dit anteriorment en el document, integrar *CkEditor* dins qualsevol pàgina web és una tasca senzilla. Només cal disposar de l'editor, que es pot descarregar a la pàgina oficial de *CkEditor* de manera gratuïta, i incloure'l dins del codi del projecte. En el moment que estigui inclòs caldrà instanciar-lo dins un element *HTML textarea* mitjançant un *script* de *javascript* amb la funció `CKEDITOR.replace('editor')` on `'editor'` és el *id* del *textarea* on es vol incloure *CkEditor*. A partir d'aquest moment *CkEditor* ja estaria integrat dins de la pàgina web.

*CkEditor* proporciona moltes possibilitats a l'hora de configurar l'editor i adaptar-lo al disseny i les necessitats de cada usuari. Permet modificar completament el seu disseny integrant *css* per tots els apartats de l'aplicació, les seves característiques o el llenguatge amb el qual *CkEditor* es comunicarà amb l'usuari. En el cas particular d'aquest projecte només s'ha necessitat modificar l'amplada i l'alçada d'aquest per adaptar-lo a la pàgina web modificant els paràmetres de configuració `CKEDITOR.config.height` i `CKEDITOR.config.width`.

## 9.2 Generació del *plugin* QuizEditor

El primer pas per desenvolupar un plugin dins l'editor és incloure dins el directori plugins, inclos al directori arrel de CkEditor, una carpeta amb el nom del plugin que es vol generar, en aquest cas, quizeditor (Fig. 9.1).

Dins el directori s'afegirà el fitxer *plugin.js* on estarà la lògica del *plugin*, els fitxers addicionals que es necessitin per el desenvolupament de l'aplicació i l'icona que tindrà el projecte dins els subdirectori *icons*.

Quan s'hagi creat l'estructura on s'emplaçaran tots els fitxers del plugin es començarà a desenvolupar la lògica dins *plugin.js*. El primer que caldrà fer es instanciar el botó que executarà l'aplicació e incloure en ell la icona que es veurà en l'editor i el codi que executarà. Per realitzar això s'utilitzarà la comanda següent:

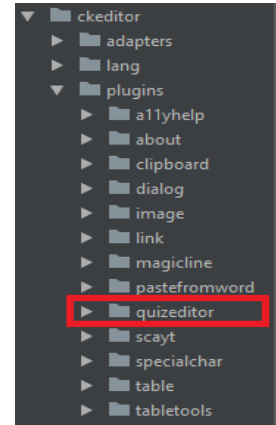


Fig.9.1 integració del directori

```
CKEDITOR.plugins.add( 'quizeditor', {  
    init: function( editor ) {  
  
        ...  
    }  
}
```

El primer paràmetre de la funció serà el nom que el *plugin* tindrà, a continuació s'inclouran els paràmetres de configuració del plugin. Dins del paràmetre *init* estarà definida la lògica del plugin. Es definiran les variables d'entorn que siguin necessàries, es carregaran totes les classes que el plugin utilitzarà i s'instanciaran els botons que es necessitin.

Per carregar els fitxers que el plugin necessitarà s'utilitzarà la funció:

```
CKEDITOR.scriptLoader.load(CKEDITOR.getUrl( path ) )
```

On path és la direcció on es troba el fitxer que s'ha d'inserir.

Per carregar els botons dins de la lògica del *plugin* CkEditor ofereix la comanda:

```
editor.ui.addButton( 'editQuiz',  
    {  
        label: 'missatge',  
        icon: 'icona',  
        click: ...  
    }  
)
```

El primer paràmetre de la funció serà el nom del botó que s'està afegint, que servirà per poder referenciar-lo posteriorment. En el segon paràmetre s'inclourà la configuració del

botó. A *label* s'inclourà el text informatiu del botó, a *icon* la direcció on es troba la icona que utilitzarà i a *click* el codi que s'executarà quan es faci clic sobre el botó.

Per defecte CkEditor no inclou tots els plugins que s'incorporen al directori plugins, si no que es necessari afegir-lo juntament amb els paràmetres de configuració, per fer-ho cal incloure la línia de codi següent juntament amb els paràmetres de configuració:

```
CKEDITOR.config.extraPlugins = 'quizeditor';
```

A partir d'aquest moment el plugin ja estaria inclòs dins l'editor integrat al nostre web (Fig. 13.1).

## CKEditor plugin per generar mapes d'imatges

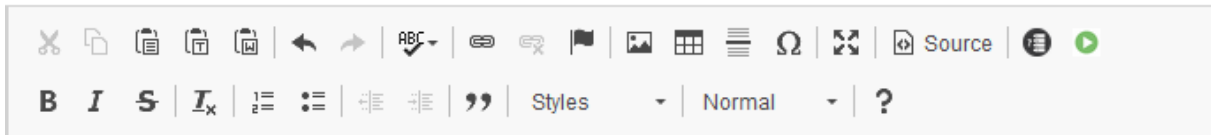


Fig. 9.2 Representació de la barra d'eines del web amb CkEditor i el plugin integrats

## 9.2 Botó d'edició d'activitats

En el moment en el que el *plugin* està integrat dins l'editor cal començar a desenvolupar el codi que aquest executarà. Com hem dit anteriorment aquest codi es realitza dins el paràmetre *click* de la funció *CKEDITOR.plugins.add*. El codi implementat és el següent:

```
editor.ui.addButton( 'editQuiz',  
  {  
    label: 'Crea la teva activitat',  
    icon: path + '/icons/quizeditor.png',  
    click: function(editor) {  
      var selected = editor.getSelection().getSelectedElement();  
      if(selected && selected.is("img")) {  
        generateEditorDialog(editor, selected);  
      }else{  
        alert(infoErrorEditor);  
      }  
    }  
  }  
);
```

Aquesta secció de codi està inicialitzant el botó *editQuiz* afegint-li un missatge informatiu i una icona. Dins del paràmetre *click* s'està comprovant si l'usuari ha seleccionat un element en l'editor i si aquest és una imatge. Si això és així executa la funció *generateEditorDialog*, de la que parlarem més endavant. Si no, mostra un missatge d'error.

Tan *path* com *infoErrorEditor* son variables globals prèviament definides.

## 9.3 Creació de la pantalla d'edició d'activitats

### Problemes durant en desenvolupament del projecte

És important comentar que en aquest moment del desenvolupament del projecte es va arribar a un punt d'inflexió important que repercutiria en el transcurs del projecte.

En un primer moment es va intentar generar la pantalla d'edició d'activitats utilitzant els diàlegs que proporciona *CkEditor*, i encara que en un principi va ser molt costós realitzar algunes de les funcionalitats dins d'aquests diàlegs poc a poc s'aconseguien resultats. Però es va arribar a un punt on les funcions que proporcionava la *API de CkEditor* per realitzar aquests diàlegs, no proporcionaven les funcionalitats que es necessitaven per el correcte desenvolupament del projecte, i d'aquesta manera tot el procés de creació de la pantalla d'edició es va haver de refer des de l'inici, invalidant així varies setmanes de desenvolupament del projecte. Es va trobar una solució utilitzant els *modals* de *jQuery* per crear internament el diàleg d'edició d'activitats sense necessitat d'utilitzar el diàlegs preestablerts per *CkEditor*.

Aquest fet ha suposat un problema per el correcte seguiment de la planificació temporal del projecte i ha suposat que algunes funcionalitats s'hagin eliminat per poder arribar a temps a la entrega del projecte.

### Solució del problema utilitzant *modals* de *jQuery*

En el moment en que un usuari tingui una imatge seleccionada i faci clic sobre el botó d'edició d'activitats, aquest cridarà a la funció *generateEditorDialog*, que necessitarà de dos paràmetres. Per una banda, l'instància de *CkEditor*, per poder enviar-lo al manager posteriorment, i la imatge seleccionada, per incloure-la dins la pantalla d'edició. El codi de la funció és el següent:

```
function generateEditorDialog(editor, selectedImage) {
    $.modal('contingut HTML de l'editor d'activitats');
    manager=new Object();
    initManager(manager);

    manager.load(editor, selectedImage);

    manager.draw();
}
```

Primerament, la funció *generateEditorDialog* generarà el disseny de la pantalla d'edició d'activitats utilitzant HTML, és a dir, amb un conjunt de *divs*, *labels*, *buttons* i *inputs* es crearà l'estructura que contindrà la imatge per una posterior edició. Aquest element HTML es guardarà dins un modal de *jQuery*. I aquesta variable *modal* que ens proporciona la llibreria *simpleModal* de *jQuery*, ens servirà per poder generar un *dialog* i mostrar-lo sobre l'editor. Gràcies a la utilització d'aquesta llibreria en comptes dels *dialogs* que *CkEditor* proporciona ens permet poder utilitzar les funcionalitats de *jQuery*

sobre qualsevol element generat dins el *modal*. I això amb el fet d'estar constantment pintant sobre un element *canvas* facilita molt gratament el desenvolupament del projecte.

Després de carregar el *modal* s'inicialitza l'objecte *manager* amb la funció *initManager* implementada al *manager.js*. Aquesta funció s'encarrega d'inicialitzar les variables i les funcions de les que disposarà la classe *manager*.

En el moment d'inicialitzar el *manager* ja es pot començar a treballar amb ell. Llavors precedirem a carregar la imatge dins l'editor amb la funció *load* i a fer el primer pintat del *canvas*. En aquest instant, si és la primera vegada que s'inicialitza la imatge només es pintarà aquesta dins el *canvas*, si existís una activitat prèviament editada es pintarien els seus components.

## 9.4 Manager.js

L'objecte *manager* és l'encarregat de comunicar-se amb totes les altres classes de l'aplicació. És la responsable d'escoltar tots els events que arriben des de la pantalla d'edició d'activitats, de dibuixar sobre la imatge en el moment que l'usuari clica sobre ella amb una eina de dibuix seleccionada. És també la responsable de gestionar i emmagatzemar la informació dels diferents components i mapes que componen l'aplicació.

### 9.4.1 Variables

L'objecte Manager realitza una funció molt important dins el sistema, i disposa d'un conjunt de variables d'entorn necessàries per la configuració del projecte. És important entendre la funcionalitat d'alguna d'elles.

- **img**: contindrà la imatge que s'utilitzarà per dibuixar sobre el canvas de la pantalla d'edició, aquesta imatge tindrà el tamany que necessita l'editor i no contindrà la informació dels mapes que aquesta té associats o d'algun altre paràmetre d'estil que hagi incorporat l'usuari. Tota aquesta informació es trobarà a *actualImage*.
- **actualImage**: imatge original rebuda pel manager amb tota la informació que aquesta conté.
- **sizeProportion**: paràmetre que conté la proporció d'amplada i alçada entre la imatge original i la imatge que és mostra en la pantalla d'edició. Aquesta variable s'utilitzarà per adequar la imatge al requadre de la pantalla d'edició.
- **context**: variable que guarda el context de la imatge per tal de poder dibuixar sobre ella.
- **editor**: variable d'entorn per poder utilitzar les funcionalitats de *CkEditor* dins el manager.
- **components[]**: llista d'objectes *component* que representen les zones marcades de la imatge.
- **selectedMaps[]**: llista amb les zones de la imatge que estan seleccionades en un moment determinat de l'acció d'edició.
- **isMultySectedEnabled**: variable *booleana* que indica si està activada l'opció de seleccionar diversos mapes.
- **isAlreadyAMap**: variable *booleana* que indica si existeix un mapa prèviament creat a la imatge.
- **drawTool**: indica el nom de l'eina de dibuix seleccionada. Aquesta variable pot tenir diferents estats, depenen de l'eina de dibuix escollida, si està esperant ser guardada o si no hi ha ninguna eina de dibuix seleccionada. Inicialment declarada a *none*.
- **questionFixed**: conté el text que l'usuari ha marcat com a pregunta fixada de l'exercici.
- **isActivity**: variable *booleana* que indica si el manager s'està executant en mode edició o en mode activitat.

## 9.4.2 Càrrega de l'editor

En el moment en que el manager esta iniciat, passa a estar llest per poder fer la càrrega del canvas on es deixarà la imatge llesta per poder-hi dibuixar després. Les funcions de càrrega, tant per l'editor d'activitats com per la pantalla d'execució son molt similars, i son un element important per entendre algunes funcionalitats de l'editor. En aquest apartat analitzarem el funcionament d'aquesta càrrega.

La funció de càrrega realitza el procediment següent:

1. Carrega les variables d'entorn.
2. Carrega la imatge, sobreescriu el tamany i extreu el *canvas*
3. Comprova si existeix un mapa associat.
  - 3.1 Si no existeix creem un de nou.
  - 3.2 Si existeix guardem els seus components.
    - 3.2.1 Deixem l'últim mapa seleccionat carregat.
4. Carreguem els events de ratolí.

### 1. Carrega les variables d'entorn.

```
editor=CkEditor;  
actualImage = image;  
mapsController = new Object();  
initMapController(mapsController);
```

Primer es guardan les instancies de l'objecte de *CKEditor* i la imatge passades per paràmetre i s'inicialitza el *mapController*.

### 2. Carrega de la imatge, sobreescriu la mida i extreu el canvas

```
img=new Image();  
img.src=image.getAttribute('src');  
if(img.width < img.height) sizeProportion = DIV_WIDTH /  
img.width;  
else sizeProportion = DIV_HEIGHT/ img.height;  
p.resize(img.width * sizeProportion, img.height *  
sizeProportion);  
context=$("#imageMapCanvas")[0].getContext('2d');
```

Es crea una nova imatge amb el valor de la imatge actual que s'utilitzarà posteriorment per ser dibuixada. Quan la imatge està creada es calcula la mida que hauria de tenir per que encaixes dins del *div* que la contindrà. Per no trencar la proporció de la imatge només es calcula aquest paràmetre per un dels dos costats, el més petit.

La raó d'aquest procediment és per no tenir espais blancs dins la capsa que conté el dibuix, així si una imatge és menor que el *div* que la conté s'expandirà per omplir-lo, i l'altre, que és més gran s'expandirà proporcionalment. Donat que el *DIV\_WIDTH* i el *DIV\_HEIGHT*, no tenen el mateix valor es pot donar el cas que quedés algun espai en blanc, però amb poca probabilitat.



### 3. Comprovar si existeix un mapa associat.

```
mapName = image.getAttribute('usemap');
if (mapName==undefined) {
    isAlreadyAMap=false;
    var src = image.getAttribute('src').split('/');
    mapName = src[src.length-1];

    map=new CKEDITOR.dom.element( 'map' );
    map.setAttribute('name', mapName);
}
else { ...
```

Primer es comprova si existeix un mapa associat amb la imatge, si no existeix el creem i per fer-ho agafem l'últim valor que conté la *url* d'una imatge, que en molts casos es el valor més identificatiu d'aquesta, i creem el mapa, que es guardarà dins un paràmetre global de l'aplicació.

```
else {
    mapName=mapName.replace("#","");
    var maps = editor.document.$.getElementsByName("map");
    for (i = 0; i < maps.length; i++) {
        if (maps[i].getAttribute('name')==mapName) {
            map = maps[i];
            for (ii = 0; ii < map.childNodes.length; ii++){
                if (map.childNodes[ii].getAttribute('alt') == 'mapInfo') {
                    mapsController.loadInfoInputs(
                        map.childNodes[ii].getAttribute('title'));
                }
                else{
                    var item = map.childNodes[ii].getAttribute('item');
                    var shape = map.childNodes[ii].getAttribute('shape');
                    var coords = map.childNodes[ii].getAttribute('coords');

                    var comp = new Object();
                    initComponents(comp, shape);
                    comp.setCoords(coords.split(','));
                    comp.setItem(item);

                    components.push(comp);
                    p.deNormalizeCoords(ii);
                }
            }
            if(components.length > 0) p.select(components.length-1);
            isAlreadyAMap = true;
        }
        else isAlreadyAMap = false;
    }
}
```

En aquest moment de l'aplicació, com sabem que la imatge té un mapa associat busquem entre tots els mapes que *CkEditor* troba dins de l'editor. Si no el trobes activem la variable *isAlreadyAMap* a *false*, informació que necessitarà per guardar el mapa posteriorment.

En el moment que trobem el mapa, ens ocupem d'extreure tota la informació dels mapes. Primer cal extreure la informació dels mapes amb la funció *loadInfoInputs*. Aquesta funció s'encarrega de carregar la informació dels mapes, emmagatzemada dins d'un *area* auxiliar dins el *map*. D'aquesta funció parlarem més endavant.

Tots els altres elements del *map* és carreguen dins un objecte component, s'afegeix els seus elements *item*, *shape* i *coords*. Utilitzant els paràmetres es normalitzen les coordenades per quadrar-les dins la imatge a la que anteriorment hem modificat la seva mida.

Quan tots els *components* estan carregats es deixa seleccionat l'últim element de la imatge.

#### 4. Carreguem els events de ratolí

En el moment en que el canvas s'ha carregat, i si prèviament existia un *map* associat a la imatge s'han carregat tots els seus elements, és el moment de carregar sobre el *canvas* tots els events de ratolí. Això es realitza utilitzant la funció *loadMouseEvents*.

Per entendre el funcionament dels events de ratolí sobre el teclat és important explicar correctament el funcionament de la variable *drawTool*.

Si observem el diagrama d'estats que es segueix per dibuixar una zona de la imatge (Fig. 9.3) veiem que inicialment està esperant una interacció del sistema, és a dir que l'usuari indiqui quina és l'eina que vol utilitzar per dibuixar. En el moment que un usuari esculli una de les tres eines de dibuix que proporciona l'editor (cercle, rectangle o polígon) esperarà que l'usuari comenci a dibuixar sobre la imatge.

En el moment en el qual l'usuari faci un clic amb el ratolí sobre la imatge s'iniciarà alguna de les funcions de dibuix d'aquesta. Si l'eina seleccionada és un cercle o un rectangle, es crearà l'objecte *component*, el dibuix es completarà i el sistema esperarà que l'usuari guardi (o esborri) el mapa que acaba de crear.

En l'altre cas, si l'eina seleccionada és un polígon, poden passar dues coses, que sigui la primera vegada que s'inicialitza el polígon, si això succeeix així, es crearà l'objecte *component* i el sistema esperarà que l'usuari finalitzi el dibuix. Pel contrari, si no és la primera vegada que es prem sobre la imatge es comprovarà si l'usuari està tancant o no el dibuix, si el tanca, es finalitzarà l'edició del dibuix i s'assignarà a aquest *component* a l'atribut *shape* com a *poly*, en comptés de *polygon* i es tornarà a l'estat en el que la variable *drawingTool* espera una interacció del sistema.

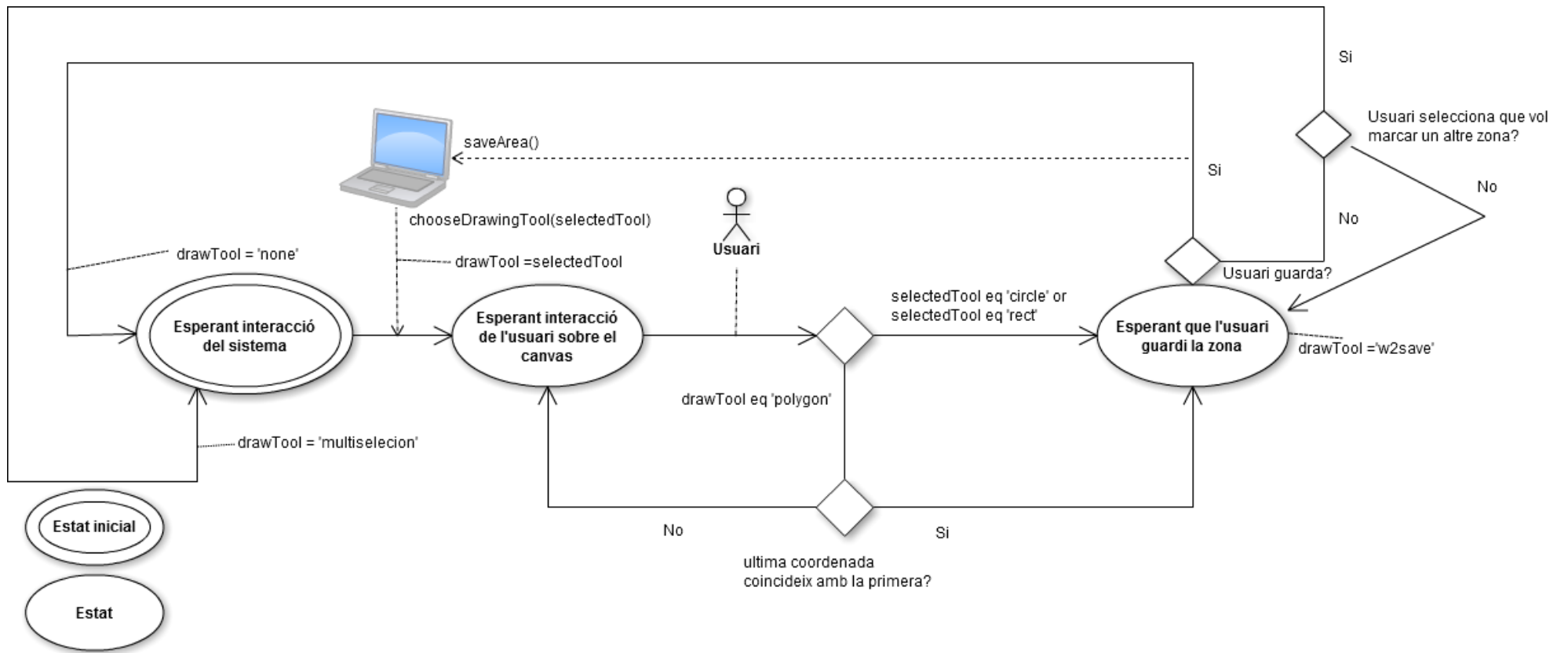


Fig. 9.3 Diagrama d'estats per dibuixar una zona de la imatge

Després d'explicar la lògica de la variable *drawingTool*, és més entenedora la funció *loadMouseEvents* del *manager*.

Primer s'assigna l'event de fer un clic amb el ratolí (*mousedown event*).

```
$("#imageMapCanvas").mousedown(function(e) {
    mouseX = e.pageX-$("#imageMapCanvas").offset().left;
    mouseY = e.pageY-$("#imageMapCanvas").offset().top;
    mousePressed=true;
});
```

El primer pas és assignar les variables de posició del ratolí dins el canvas, per això és necessari restar-li l'*offset* que el *canvas* aporta. També s'activa la variable *mousePressed* que indica si el ratolí està pressionat.

```
switch(drawTool) {
    case 'none':
        for(var i=0; i<components.length; i++) {
            if(isAMapClicked(i)) {
                p.select(i);
            }
        }
        break;
}
```

El següent es veure que conté la variable *drawTool*. Si no conté ninguna eina de dibuix és mira si la posició clicada coincideix amb algun *component* de la imatge, si es així aquest es selecciona.

```
case 'polygon':
    if(components.length == 0 ||
        components[components.length-1].getShape() != 'polygon')
        p.polyInitialize();
    else p.polyClick();
    break;
```

Si l'eina de dibuix és *polygon*, es comprova si és la primera vegada que es fa clic sobre la imatge, mirant si l'últim element de la llista de components és un *polygon*. Si ho és, com aquest estat només existeix en el moment en què s'està dibuixant significa que no és la primera coordenada de la imatge, sinó ho és, sí que caldrà inicialitzar aquest component.

```
case 'rect':
    p.squareClick();
    break;
case 'circle':
    p.circleClick();
    break;
}
```

Si l'eina dibuix és un rectangle o un cercle, només cal dibuixar-ho directament, ja que en aquest cas, el dibuix queda pendent de guardar, però es pot modificar arrastrant el ratolí sobre alguna coordenada de la zona marcada.

Finalment només cal cridar a la funció `p.draw()` encarregada de dibuixar sobre la imatge.

```
$("#imageMapCanvas").mouseup(function(e) {  
    mousePressed=false;  
});
```

Si es deixa de fer clic amb el ratolí es desactiva la variable `mousePressed`.

```
$("#imageMapCanvas").mousemove(function(evt) {  
    prevMouseX=mouseX;  
    prevMouseY=mouseY;  
    mouseX=evt.pageX-$("#imageMapCanvas").offset().left;  
    mouseY=evt.pageY-$("#imageMapCanvas").offset().top;  
    if(mousePressed) {  
        p.mouseDragged();  
        p.draw();  
    }  
});
```

Finalment l'event de moure el ratolí sobre la imatge rep la posició del ratolí i si aquest està pressionat, s'activa la funció de dibuix `mouseDragged` i es dibuixa sobre la imatge.

Aquestes funcions de dibuix és comentaran posteriorment.

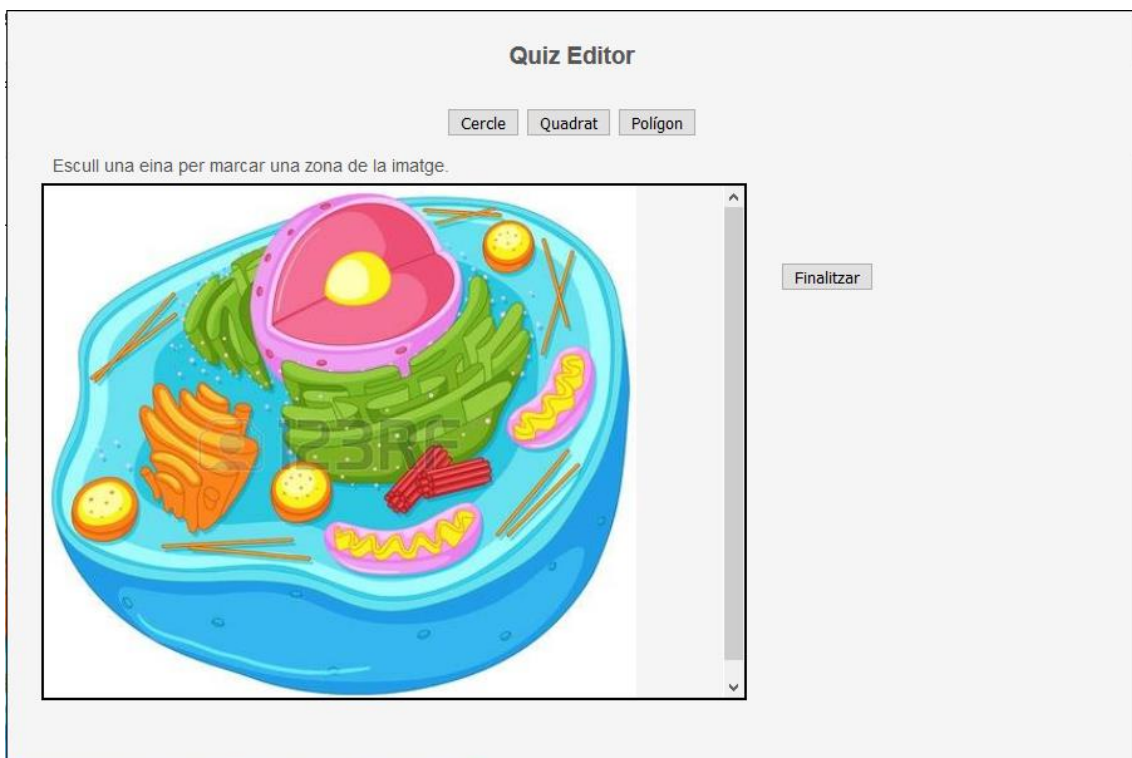


Fig 9.4 Pantalla inicial d'edició d'activitats

### 9.4.3 Càrrega de l'activitat

La funció encarregada de carregar els elements del manager en executar una activitat, *loadActivity*, és molt semblant a la funció *load* comentada en l'apartat anterior. L'única diferència que tenen és que aquesta funció sap que la imatge que està rebent ja té incorporada un mapa amb zones marcades per QuizEditor. És per això que realitza exactament el mateix, però sense haver de fer comprovacions sobre el mapa, i a part activa la variable *isActivity* per funcions de dibuix posteriors.

### 9.4.4 Funcions de pintat del *canvas*

#### Generar components

La classe *manager* realitza moltes funcionalitats essencials dins la lògica del projecte i entre elles es troben les funcions encarregades de mostrar el procés d'edició que està realitzant un usuari sobre una imatge. Per fer-ho utilitza les funcions ofertes pel *context* de *canvas* que permet dibuixar sobre un element HTML *canvas*.

En el projecte s'ofereixen tres eines de dibuix, el cercle, el rectangle i el polígon. Aquests elements es transformaran posteriorment en les àrees d'un *map*.

Un *map* és un element HTML associat a una imatge concreta que proporciona la informació de les zones clicables de la imatge. Aquestes àrees, representades com a elements *area* que penjen d'un *map* tenen dos atributs importants per la nostra aplicació. Les coordenades (*coords*) que emmagatzemen la informació dels punts de la imatge que contenen una zona clicable, la forma de l'àrea (*shape*) que informen del tipus d'element *area* i serveixen per donar forma a les diferents coordenades.

Un *area* pot ser representada de tres formes diferents, depenen del seu atribut *shape*:

- Per un rectangle, el valor de l'atribut *shape* serà *rect* i es representarà amb dues coordenades, una per la cantonada superior esquerra, i l'altre per la cantonada superior dreta.
- Per un cercle, el valor de l'atribut *shape* serà *circle*, i es representarà amb una coordenada, que serà el punt central del cercle, i amb el valor del radi d'aquest.
- Per un polígon, el valor de l'atribut *shape* serà *poly*, i es representarà amb una coordenada per cada cantonada de l'element.

Seguint aquesta idea s'han utilitzat diverses funcions per crear aquests elements i gestionar el valor del paràmetre *drawTool*, aquestes funcions són *circleClick*, *squareClick*, *polyClick*, *polyInitialize* i *finishDrawing*.

Les funcions *circleClick* i *squareClick* inicialitzen un component indicant-li el tipus de la figura que s'està dibuixant i li afegixen les coordenades inicials. En el cas del cercle és genera inicialment amb un radi 10 agafant com a coordenada central el punt on clica l'usuari. Si el que es dibuixa és un rectangle, es crea una figura tenint en compte el punt que indica l'usuari com a primer punt i es generà un quadrat de 10x10 inicialitzant el

segon punt 10 píxels enredra de les x i 10 de les y (dins de la funció *addCoords* es comprova que el valor de les coordenades no sigui inferior a 0). Quan el component està generat s'indica al programa que el dibuix està finalitzat.

```
p.circleClick = function(){
    var comp = new Object();
    initComponents(comp, "circle");
    comp.addCoords(mouseX, mouseY); comp.addCoord(10);
    components[components.length] = comp;
    p.finishDrawing('circle');
};

p.squareClick = function(){
    var comp = new Object();
    initComponents(comp, "rect");
    comp.addCoords(mouseX, mouseY);
    comp.addCoords(mouseX-10, mouseY-10);
    components[components.length] = comp;

    p.finishDrawing('rect');
};
```

De la mateixa manera, al generar un polígon es crea l'element *component* i se li afegeixen les coordenades. Quan l'usuari segueix marcant punts de la imatge es crida a la funció *polyClick* que comprova si l'última coordenada marcada per l'usuari està relativament a prop de la primera. Si és així finalitza el component.

```
p.polyInitialize = function(){
    var comp = new Object();
    initComponents(comp, "polygon");
    comp.addCoords(mouseX, mouseY);
    components[components.length] = comp;
    p.select(components.length-1);
};

p.polyClick = function(){
    var actualComp = components[components.length - 1];
    if(isNear(mouseX, mouseY, actualComp.getCoord(0),
        actualComp.getCoord(1)))
        p.finishDrawing('poly');
    else components[components.length -
1].addCoords(mouseX, mouseY);
};
```

Quan un element està acabat es crida la funció *finishDrawing* que finalitza l'última zona marcada per l'usuari, selecciona el component marcat, canvia el panell d'informació i canvia la variable *drawTool* a l'estat d'espera a què l'usuari guardi una zona.







A l'hora de dibuixar l'element de tipus rectangle s'utilitza la funció *strokeRect* definida pel *context*. Aquesta rep com a paràmetre la cantonada superior esquerra del rectangle i els paràmetres d'amplada i alçada. Aquest primer paràmetre està emmagatzemant dins el *component* a dibuixar, però en canvi l'alçada i l'amplada esdevenen de restar-li els punts *x* e *y* a la cantonada inferior dreta.

L'últim element que pot arribar per dibuixar és el polígon, aquest pot estar en dues fases, una quan l'usuari encara l'està dibuixant, on aquest polígon no estarà completat, i l'altre quan ja ha finalitzat, en aquest cas l'últim punt de l'element s'uneix amb el primer.

S'ha utilitzat una funció *drawPolygon* encarregada de dibuixar l'element. Aquesta utilitza les funcions de context per dibuixar la traça punt a punt. Primer s'inicialitza la traça que es vol dibuixar amb la funció de context *beginPath*, posteriorment es mou la traça fins al primer punt, i s'itera sobre totes les coordenades dibuixant una línia entre elles. En aquest moment depenent de si és o no un polígon, és a dir, si el seu atribut *shape* és *polygon* (inacabat) o *poly* (acabat) finalitzat es tanca la traça amb la funció *closePath*. Finalment es representa el dibuix dins del *canvas* utilitzant la funció *stroke*.

```
p.drawPolygon = function(i) {
    context.beginPath();
    context.moveTo(components[i].getCoord(0),
                    components[i].getCoord(1));
    for(var j=2; j<components[i].getCoords().length; j=j+2) {
        context.lineTo(components[i].getCoord(j),
                       components[i].getCoord(j+1));
    }
};
```

Un polígon acabat i un d'inacabat tenen les mateixes coordenades, però es veuen de diferent manera, ja que es representen amb un *shape* diferent. I com hem vist a l'hora de dibuixar-lo no s'acaba l'última traça.



Fig. 9.6 Representació d'un component de tipus *polygon* (esquerra) i un altre de tipus *poly* (dreta)

## 9.4.5 Funcions per guardar i carregar dades

Com s'ha explicat durant el document les zones que marca un usuari es guarden en dues bandes, la classe component s'encarrega d'emmagatzemar les dades de posició de cada zona, les coordenades que l'ocupen, la forma que té i un identificador al mapa que conté la informació de l'activitat de pregunta-resposta associada a ella. I la classe *mapObject* s'encarrega de guardar aquesta informació de pregunta-resposta.

Troblem en el projecte quatre moments en què es realitza un procés de guardat o carregat de dades. En el moment en què un usuari guarda una zona de la imatge es realitza un procés d'emmagatzematge local en el qual aquestes dades es mantenen dins les classes del programa. Si el mateix usuari, dins la pantalla d'edició o durant l'execució de l'activitat necessita aquestes dades llavors el sistema les ofereix i les mostra d'una manera o altra depenen la situació. Però en el cas que un usuari tingui intenció de finalitzar una activitat aquestes dades s'han d'emmagatzemar dins l'HTML i preparar-les per se posteriorment carregades pel *manager*, i enviades a l'executor d'activitats si fos necessari.

```
p.saveArea = function() {
    ...
    var actualMap = components[selectedMaps[0]].getItem();
    if(actualMap.indexOf("map") > -1) {
        mapsController.modifyMap(newQuestion,newAnswer,
                                isTest,correctAnswer,actualMap);
    }
    else{
        var nMap = mapsController.addMap(newQuestion,newAnswer,
                                        isTest,correctAnswer);

        for(i = 0; i < selectedMaps.length; ++i) {
            components[selectedMaps[i]].setItem("map" + nMap);
        }
    }
    drawTool = 'none';
    selectedMaps = [];
}
```

Quan un usuari indica que vol guardar una zona el sistema comprova si aquesta zona està sent creada, llavors genera un *mapObject* que associarà a l'element (o elements) a marcar, o editada, en aquest moment buscaria quin és l'element *mapObject* associat amb ell i el modifica. Posteriorment s'eliminen els mapes de la llista de seleccionats i la variable *drawTool* torna a l'estat inicial (Veure Fig. 9.3).

Perquè el sistema pugui emmagatzemar les dades dins de l'HTML de l'editor i aquesta informació pugui ser posteriorment emmagatzemada per l'usuari quan vulgui enviar-la a la seva base de dades s'ha ideat una estructura de dades que estarà inclosa dins un element *area* que penjarà conjuntament amb les altres zones de la imatge del *map*.

Aquestes dades es guardaran amb l'estructura següent:

```
mapid@@pregunta##resposta1##esTest:respostaTest:estatPregunta||mapl...
```

En aquesta estructura **mapid** representà l'identificador de pregunta-resposta associat als components de la imatge. **pregunta** i **resposta** fan referència als paràmetres que una determinada zona té com a pregunta i resposta. **esTest** és un paràmetre de configuració que indica amb un 1 que la pregunta indicada és un test, i amb un 0 que no ho és, **respostaTest** és el valor numèric de la resposta tipus test, que pot anar-hi entre 1 i 4, si la pregunta no és de tipus test, aquest valor serà un 0. Finalment **estatPregunta** és valor que s'utilitza durant l'execució de l'activitat que indica si la pregunta no ha estat resposta amb un 0, si ho ha estat correctament amb un 1 i si és incorrecta amb un 2. Cada valor es separa de l'anterior amb les dues barres verticals "||".

Aquesta informació s'extreu en el moment en què l'usuari finalitza l'edició d'una imatge amb la funció *extractMapsInfo* del *mapController*, i és càrrega a l'obrir amb l'editor una imatge prèviament editada, o en carregar la imatge des de l'activitat amb la funció *loadInfoInputs*.

Un possible exemple d'aquest element seria el següent:

```
<area item="" href="" alt="mapInfo" title="map1@@Indica la part de la cel·lula: ##mitochondria##0:0:0||map2@@ Indica la part de la cel·lula: ##nucli##0:0:0||map3@@Indica la part de la cel·lula: ##ribosomes;;centriolo##1:2:0"/>
```

## 9.5 Altres classes del projecte

En el document s'han explicat les funcions principals que componen l'objecte *manager*, però també és important comentar el funcionament de les altres classes que conté el projecte.

Per una banda trobem la classe *component* que és l'encarregada d'emmagatzemar la informació de les àrees que representen una zona marcada de la imatge. Aquesta classe contindrà les variables que componen aquest objecte: *title*, *href*, *alt*, *coords* e *item*. Les variables que tenen un significat dins el projecte són *coords* que contindran la informació de les coordenades del *component* dins la imatge, e *item* que guardarà el nom del *mapObject* que té associat.

Troblem també les classes *mapObject* i *mapController*, aquestes són les encarregades d'emmagatzemar la informació pertinent de cada zona o conjunt de zones marcades a la imatge. Un *mapObject* conté les variables que representen aquesta informació, la pregunta, la resposta, els paràmetres de configuració de les preguntes tipus test, i l'estat que indica si la pregunta és o no correcta.

El *mapController* per una banda conté i gestiona els diferents elements *mapObject* i per l'altre coneix tots els paràmetres de configuració d'una activitat concreta, el nombre

d'errors que s'han comés, el nombre de preguntes totals, les respostes i les que falten per respondre. D'aquesta manera durant l'execució de l'activitat, a través del manager, es realitzaran les crides necessàries al controlador per rebre la informació que es mostrarà a l'usuari per pantalla.

## 9.8 Generació de l'apartat d'execució d'activitats

En el moment que un usuari generà una activitat utilitzant QuizEditor i la guarda dins una imatge, aquesta estarà disponible per ser executada. Per tal d'executar aquesta imatge s'ha decidit incloure un segon botó que tingui la funcionalitat d'obrir una nova pàgina on s'executarà l'activitat.

Per fer-ho caldrà instanciar un segon botó amb la comanda `addButton` que ofereix CkEditor.

```
editor.ui.addButton( 'executeQuiz',
    {
        label: 'Executa la activitat',
        icon: path + '/icons/play-icon.jpeg',
        click: function(editor) {
            var selected =
                editor.getSelection().getSelectedElement();
            if(selected && selected.is("img")) {
                var useMap = selected.getAttribute('useMap');
                if(useMap == null) alert(infoErrorExecution);
                else { ... }
            }
            else {
                alert(infoErrorEditor);
            }
        });
```

Per instanciar aquest segon botó es tornen a incloure els paràmetres *label* e *icon* de la mateixa manera que s'havia fet amb el primer. Dins del paràmetre *click* es defineix la funció que s'executarà al fer clic sobre aquest.

Primerament es comprovarà que s'estigui seleccionat una imatge dins l'editor i que aquesta contingui un element *map* associat a ella, aquest element es troba referenciat dins l'atribut *useMap*.

Quan es troba el nom de l'element *map* associat a la imatge es busca aquest element, i la seva informació es guarda dins una variable per poder ser posteriorment referenciat.

```
var maps = editor.document.$.getElementsByTagName("map");
var htmlMap;
for (i = 0; i < maps.length; i++) {
    if (maps[i].getAttribute('name') == useMap.replace("#", "")) {
        var map = maps[i];
        htmlMap = map.outerHTML;
    }
}
```

Per acabar, el botó d'execució d'activitats obre en una nova pestanya la vista *Activity.html* on es troba el procés que seguirà la futura activitat. Les dades de la imatge seleccionada i el *map* associat a ella, el qual conté no només els elements seleccionables de la imatge, sinó també la informació de tots ells. Per fer-ho obrim una nova finestra amb la funció *window.open* on passem per paràmetre el *path* de la vista que volem obrir y com volem obrir aquesta nova finestra, indicant "blank" s'obrirà en una nova pestanya. Aquesta funció retorna un element que fa referència a aquesta nova finestra, llavors en ell indiquem que quan la finestra estigui carregada insereixi un element *div* on es trobarà la nostra imatge i el seu mapa associat.

```
var src = selected.getAttribute('src');
var openWindow = window.open(path + "activity.html" , "_blank");
$(openWindow).ready(function() {
    $(openWindow.document.body).append('<div
        id="imageMapCanvasDiv" style="position:relative;"></div>');
    $(openWindow.document.body).append(htmlMap);
});
```

## 9.9 Procés d'execució d'activitats

El plugin es l'encarregat de generar la pantalla on s'executarà l'activitat prèviament definida però és dins *Activity.html* on es troba la lògica del programa.

Per defecte *Activity.html* conté únicament un text i un botó, inicialment la imatge està inserida amb opacitat 0. En el text s'informa a l'usuari que per iniciar l'activitat ha de fer clic sobre el botó, i aquest s'encarrega de cridar a la funció *startActivity* definida en el fitxer.

**L'activitat ha estat carregada correctament.**

**Per iniciar-la només has de clicar el botó. Molta sort!**

Iniciar activitat

Fig. 9.7 Estat inicial de la pantalla d'execució d'activitats

El primer que realitza la funció *startActivity* és mostrar la informació que li serà útil a l'usuari durant la realització de l'activitat, primer crearà un objecte *canvas* en la mateixa posició que la imatge. D'aquesta manera la imatge que té una opacitat 0 no serà visible, però les seves àrees sí que seran clicables, i com el *canvas* es troba a la mateixa posició i en ell està la imatge amb les zones dibuixades es veurà aquesta i es podran clicar cadascuna d'aquestes zones (Fig. 9.8).

0/4 respostes correctes



Fig 9.8 Pantalla d'execució d'activitats en el moment en que es carrega.

La funció *startActivity* és necessària per poder assignar a cada àrea del mapa la funcionalitat d'obrir un diàleg amb la pregunta concreta al clicar sobre ella. Per fer-ho modifica mitjançant *jQuery* l'event *onClick* de tots els elements que continguin l'atribut *item*, que son totes les àrees que te el *map* associat a la imatge.

```
$( '[item]' ).click(function () {  
    $('#answerGiven' ).val ( "" );  
    var mapId = $( this ).attr ( "item" );  
    $('#mapId' ).val ( mapId );  
}
```

Primer s'assegura que el camp de la resposta estigui buit i agafa l'atribut *item* on està emmagatzemat el *id* del *mapObject* que té associat.

Posteriorment comprova si la pregunta és o no tipus test i va incloent la informació dins dels paràmetres definits a un diàleg que es troba dins de la vista.

```
if ( manager .getIsMapATest ( mapId ) ) {  
    $('#answerGiven' ).hide ( ) ; $('#testDiv' ).show ( ) ;  
    var answers = manager .getMapAnswer ( mapId ).split ( ';;' );  
    for ( var i = 1 ; i <= 4 ; ++i ) {  
        if ( i <= answers .length ) {  
            $('#radio' + i ).show ( ) ;  
            $('#label' + i ).html ( answers [ i - 1 ] ) ;  
        }  
        else $('#radio' + i ).hide ( ) ;  
    }  
} else {  
    $('#answerGiven' ).show ( ) ;  
    $('#testDiv' ).hide ( ) ;  
}
```

En aquest moment s'obre el diàleg i es mostra la informació a l'usuari.

```
$( "#questionDialog" ).dialog({  
  title:manager.getMapQuestion(mapId),  
  width:450  
});
```

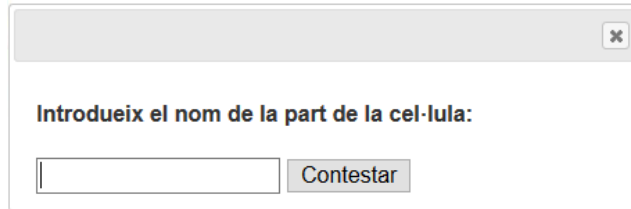


Fig. 9.9 Diàleg que apareix al seleccionar una zona amb una pregunta

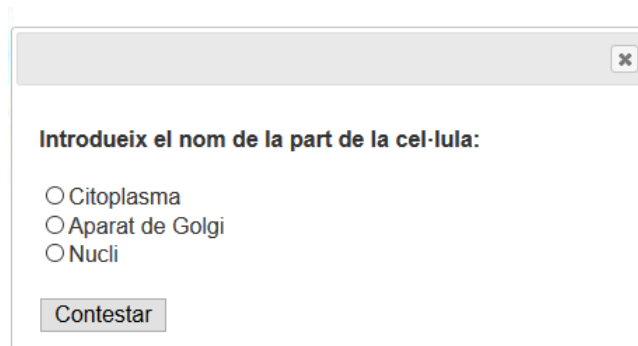


Fig. 9.9 Diàleg que apareix al seleccionar una zona amb una pregunta tipus test

Quan una d'aquestes preguntes és resposta el *manager* rep la resposta i la compara amb la que te emmagatzemada en el *mapObject* associat a aquella zona, retorna l'estat de la resposta i mostra un feedback a l'usuari indicant-li si ha respost correctament o no la pregunta i modifica la informació que veu l'usuari tenint en compte l'estat de la seva resposta.

**Resposta incorrecta!**

0/3 respostes correctes

Errors:1



**Resposta correcta!**

1/3 respostes correctes

Errors:1



Fig. 9.10 Feedback que rep l'usuari al respondre una pregunta incorrecta o correctament



## 10. *Plugins* complementaris a QuizEditor

Com s'ha comentat anteriorment en aquest document, l'eina que es proposa en el projecte, gràcies al fet d'estar inclosa dins un editor de text en línia com és *CkEditor* li permet enriquir-se de tots els avantatges que això li aporta. I un d'ells és poder complementar la seva funcionalitat gràcies a la gran quantitat de *plugins* que la comunitat de *CkEditor* ha creat per aquest editor.

Alguns *plugins* seran necessaris per donar un bon ús de l'eina QuizEditor, com permetre a l'editor pujar una imatge o guardar les dades d'aquest, hi haurà un altres que li serviran per complementar les activitats que es poden crear gràcies a l'editor.

### 10.1 *Plugin* per inserir imatges

És imprescindible per utilitzar QuizEditor que existeixi una imatge en l'editor, és per aquesta raó que incloure una eina dins l'editor que permeti inserir una imatge, ja sigui pujant-la des d'un directori personal o agafant el link d'aquesta a Internet és essencial per un correcte desenvolupament de la utilització de l'eina.

Dins el paquet bàsic de *CkEditor* ja s'inclou un *plugin* capaç d'inserir imatges des del web, però és possible que algun usuari vulgui alguna eina que s'adapti millor al que necessiti.

### 10.2 *Plugin* per guardar dades

Una característica complementaria al projecte presentat, és el fet de poder guardar les dades de l'activitat creada. *CkEditor* proporciona una funcionalitat bàsica per realitzar-ho, si l'editor s'inclou dins d'un formulari tot el text HTML que es trobi en el seu interior serà un paràmetre de l'àrea de text on es trobi.

És probable que a algun usuari no li interessi guardar les dades del seu projecte en cap base de dades, perquè no disposa d'ella o perquè el funcionament de la seva activitat no ho requereix. És per això que existeixen *plugins* per extreure les dades de l'editor i descarregar-les en un fitxer per poder carregar-les posteriorment.

### 10.3 Altres *plugins* d'edició d'activitats

Si un usuari té la intenció de realitzar una activitat interactiva utilitzant QuizEditor, o disposa d'un web que permet crear activitats interactives utilitzant *CkEditor* i vol incloure QuizEditor és molt recomanable que utilitzi més *plugins* que permetin crear activitats interactives diferents per tal de poder donar una millor experiència a l'usuari final.

# 11. Sostenibilitat i compromís social

Per avaluar la sostenibilitat i el compromís social del projecte s'han d'analitzar tres àmbits importants, l'impacte ambiental, l'econòmic i el social.

## 11.1 Ambiental

Si ens centrem en la sostenibilitat ambiental, podem veure que aquest projecte té un impacte positiu a la societat. Podem basar aquesta argumentació en dos pilars.

Per una banda el disseny del projecte no tindrà un impacte ambiental significatiu, ja que els recursos materials que es necessiten per dur-lo a terme són escassos i reutilitzables.

Per l'altra banda el fet que les escoles utilitzin el producte que es planteja en aquest document pot tenir un impacte ecològic, que ajudarà a fer que les escoles deixin de gastar tants fulls en realitzar activitats pels alumnes.

## 11.2 Econòmic

S'ha realitzat una aproximació del cost del projecte i es veu clarament que donat que és un projecte de caràcter social no es preveu un retorn de la inversió a curt termini. Encara que aquest projecte no sigui econòmicament viable des d'un punt de vista empresarial sí que ho és des d'un punt de vista educatiu. Les escoles no estaran obligades a pagar res per tal d'adquirir el producte i podran rebre els seus beneficis d'una forma gratuïta.

## 11.3 Social

Un dels punts forts d'aquest projecte és l'impacte social que pot arribar a tenir. El seu objectiu final és ajudar a fer que l'alumnat es pugui sentir més implicat en els estudis i afavorir al fet que pugui acabar els estudis obligatoris i seguir en el món educatiu.

En l'àmbit personal el projecte m'ha ajudat a veure les mancances en les quals es troba actualment el sistema educatiu a causa de la falta de recursos, on els més afectats són els estudiants més joves. I una de les millors formes d'atacar aquest problema és utilitzant tot el que ens aporten les noves tecnologies.

Gràcies al projecte que es presenta en aquest document s'oferirà una eina útil per la docència, i s'intentarà que sigui la més senzilla possible perquè els professors no tornin a optar per les activitats en paper. Per part de l'estudiant gràcies al feedback immediat que proporcionaran les activitats interactives realitzades amb aquesta eina, aconseguiran assimilar millor els conceptes sense la necessitat de tenir un professor sempre darrere seu.

## 11.4 Matriu de sostenibilitat

Tenint en compte el que es descriu en els apartats anteriors passem podem extreure la següent matriu de sostenibilitat.

Sostenibilitat	Ambiental	Econòmica	Social	Total
Planificació	Recursos i petjada ecològica	Viabilitat econòmica	Impacte en la societat	
Valoració	8	6	9	23

## 12. Conclusions

### 12.1 Desviació temporal final

En el desenvolupament del projecte s'han anat trobant obstacles ocasionats per la falta de coneixement del llenguatge i del funcionament de la API de *CkEditor*. Encara que molts d'ells s'han pogut solucionar molts d'aquests problemes han proporcionat un gran cost temporal dins el procés del projecte.

Algunes de les funcionalitats presentades en la planificació temporal prèvia del projecte, concretament a la tercera fase de desenvolupament s'han necessitat extreure de les tasques a implementar. En la planificació del projecte s'havia proposat realitzar una eina que representés un pinzell per marcar una zona de la imatge, i encara que es va procedir al seu desenvolupament, no va arribar a finalitzar-se. També algunes opcions de personalització de l'activitat proposades a l'inici del projecte no s'han implementat i passarien al seu desenvolupament en una etapa futura del projecte.

Els problemes comentat anteriorment no ha suposat que ninguna de les funcionalitats bàsiques del projecte quedessin fora del desenvolupament, i totes aquestes s'han pogut documentar i testejar amb profunditat. Encara que algunes de les opcions secundaries no estiguin implementades s'han complert els requisits del projecte correctament.

### 12.2 Conclusions

Al començament d'aquest document es comentava l'elevada alta taxa de fracàs escolar que afecta Espanya en comparació amb altres països europeus. És per aquesta raó que és necessari que apareguin solucions que intentin aportar una mica de llum oferint eines a la docència que puguin ajudar a captivar les ganes que tenen els seus alumnes per aprendre. Aquest projecte intenta ser una d'aquestes eines.

La realització d'aquest projecte ha servit per desenvolupar una eina que pugui donar un ajut a la societat en aquest aspecte per una banda. Però el que és més important, ha sigut una oportunitat per veure el procés que s'ha de seguir per desenvolupar un projecte real i ha ajudat l'estudiant a entendre i millorar els coneixements apresos en altres assignatures.

## 13. Bibliografía

1. España sigue liderando el ranking de abandono escolar en la UE(27/04/2016), EIMundo Recuperado de:  
<http://www.elmundo.es/sociedad/2016/04/27/5720a75622601d78798b466e.html>  
Consultado el 25/02/2017
2. Herrera, M. (2016). El valor de la escuela y el fracaso escolar. *REICE. Revista Iberoamericana sobre Calidad, Eficacia y Cambio en Educación*, 7(4),  
<https://revistas.uam.es/index.php/reice/article/view/5391/5830>  
Consultado el 25/02/2017
3. Epstein, M. L., Lazarus, A. D., Calvano, T. B., & Matthews, K. A. (2002). Immediate feedback assessment technique promotes learning and corrects inaccurate first responses. *The Psychological Record*, 52(2), 187,  
<http://search.proquest.com/openview/70b306d02ec963238feda4c18dc907d3/1?pq-origsite=gscholar&cbl=41800>  
Consultado el 27/02/2017
4. Schooler, L. J., & Anderson, J. R. (2008). The disruptive potential of immediate feedback.  
<http://repository.cmu.edu/cgi/viewcontent.cgi?article=1079&context=psychology>  
Consultado el 27/02/2017
5. Perales Palacios, F. J., & Dios Jiménez, J. D. (2002). Las ilustraciones en la enseñanza-aprendizaje de las ciencias. *Enseñanza de las Ciencias*, 20(3), 369-386.  
<https://books.google.es/books?hl=es&lr=&id=Y5Xw3FBCTBUC&oi=fnd&pg=PA17&dq=estimular+alumnos&ots=IS2W6r5sUn&sig=L2b6hTJYTnG5hv1LyLdeujr3IzY#v=onepage&q=estimular%20alumnos&f=false>  
Consultado el 25/02/2017