

Treball final de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Renovació del Simulador de la Màquina Senzilla amb tecnologies web modernes

Document: Memòria

Alumne: Bryan Pierre

Tutor: Antonio Bueno & Pere Vilà

Departament: Arquitectura i Tecnologia de Computadors

Àrea: Arquitectura i Tecnologia de Computadors

Convocatòria (mes/any): Febrer 2017

1. Introducció, Motivacions, Propòsits i Objectius del projecte	3
1.1 Introducció	3
1.2 Motivacions, Propòsits i Objectius	4
1.3 Estructura de la documentació	5
2. Estudi de viabilitat	7
3. Metodologia	8
4. Planificació	10
5. Marc de treball i conceptes previs	12
5.1 Arquitectura client-servidor	12
5.2 Serveis Web	13
5.3 Single Page Application (SPA)	14
5.4 Patró Model, View, Controller (MVC)	14
5.5 Sistema de Control de Versions	16
6. Requisits del sistema	17
6.1 Requisits funcionals	17
6.2 Requisits no funcionals	18
7. Estudis i decisions	20
7.1 Llenguatge de programació	20
7.2 Frameworks	21
7.3 Framework front-end	22
7.4 Llibreries	22
7.5 Imatges Vectorials SVG	23
7.6 Entorn de desenvolupament	24
7.7 Servidor	26
8. Anàlisi i Disseny del sistema	27
8.1 Anàlisi	27
8.1.1 Diagrames de casos d'ús	27
Avançar en el flux de dades	28
Retrocedir en el flux de dades	29
Carregar fitxer de configuració a la RAM	30
Editar memòria RAM	31
8.1.2 Fitxes de cas d'ús	32
Avançar en el flux de dades	32
Retrocedir en el flux de dades	33
Editar memòria RAM	34
Carregar fitxer de configuració a la RAM	35
Reiniciar la màquina	36
Reiniciar per defecte la màquina	36

Canviar mode de visualització normal a assemblador i viceversa	37
Canviar colors del flux de dades	37
8.2 Disseny del sistema	38
8.2.1 Diagrama de classes	38
9. Implementació i proves	42
9.1 Estructura de directoris del projecte	42
9.2 Càrrega de llibreries	44
9.3 Estructura index.html “Responsive”	46
9.4 Models i vistes de l’aplicació	49
9.5 Classes generals “app” i “ui”	52
9.6 Imatges SVG (Scalable Vector Graphics)	55
9.7 Representació de la memòria RAM	58
9.8 Consola	60
9.9 Interpretació entrada de dades i algoritme “parsejador”	61
10. Implantació i resultats	65
10.1 Implantació del Servidor Web	65
10.2 Fase beta de l’aplicació	73
11. Conclusions	74
12. Treball Futur	75
13. Bibliografia	76
14. Annexos	77
14.1 Índex de totes les imatges que apareixen en el projecte	77
14.2 Llicència sobre la qual està definit el projecte	79
15. Manual d’usuari	96
15.1 Requisits tècnics	96
15.1.1 Compatibilitats amb el navegador	96
15.1.2 Resolució del navegador	96
15.2 Funcionament de l’aplicació	97
15.2.1 Botons de control de l’aplicació	97
15.2.2 Càrrega del fitxer de configuració i edició de la memòria	97

1. Introducció, Motivacions, Propòsits i Objectius del projecte

1.1 Introducció

En el món de la docència ens trobem que en moltes situacions necessitem eines extres per facilitar l'aprenentatge dels alumnes. Sobretot a l'hora d'ensenyar temes més tècnics i difícils d'entendre.

En el nostre cas ens centrarem en l'ensenyament de la Màquina Senzilla que defineix el comportament més bàsic d'una computadora. Composta per uns registres de dades, una memòria RAM, una unitat aritmètica lògica i senyals de control, que més endavant explicarem.

Aquesta màquina senzilla es basa en part de la docència de l'assignatura d'Estructures i Tecnologies de Computadors II (ETCII) del primer curs del Grau d'Enginyeria Informàtica de la Universitat de Girona. Per això mateix, és una aplicació única i feta a mida, on no trobarem res semblant a la xarxa.

Abans de continuar hem de parlar de la primera versió d'aquesta aplicació feta l'any 1996 per en Ferran García i Pagans com el seu treball de fi de carrera. Una versió molt reeixida per l'època, ja que no comptaven amb els recursos tecnològics i d'informació que tenim avui dia. Amb el pas del temps aquesta aplicació web ha anat quedant obsoleta, ja que la tecnologia amb la qual està feta és molt poc segura i els navegadors estan deixant de donar suport.

Per això, aquest projecte és la renovació des de zero i amb noves tecnologies d'aquesta vella versió. Serà una aplicació web moderna feta majorment en Javascript i compatible amb els navegadors actuals. També solucionarem errors de l'antiga versió i afegirem noves funcionalitats a més de millorar l'aspecte visual.

La idea de començar aquest projecte va ser la necessitat d'adaptar una bona aplicació docent obsoleta cap a una aplicació que els alumnes d'informàtica poguessin utilitzar per aprendre de manera ràpida i intuïtiva la matèria de la Màquina Senzilla.

El repte del projecte, estarà a crear sense cap codi font anterior i des de zero la nova aplicació web de la Màquina Senzilla.

1.2 Motivacions, Propòsits i Objectius

La meva motivació per a realitzar aquest projecte ha estat aprendre com fer aplicacions webs des de zero i sobretot a mida, sense fer ús d'aplicacions que abstraïen part del procés de creació, com per exemple els gestors de continguts web.

A més, poder expandir els meus coneixements sobre llenguatges de programació com Javascript, d'estructura web HTML i de disseny CSS, que estan molt valorades, és un punt positiu i motivant.

Finalment saber que els futurs alumnes de la meva universitat podran utilitzar la meva aplicació, potser 20 anys més, és un sentiment que m'omple i m'agrada.

Com a objectiu vull crear una aplicació (SPA) “Single Page Application” del simulador de la Màquina Senzilla. Semblant a l'antiga versió però adaptada als dies d'ara. Que sigui mínimament “responsive” i que sigui intuïtiva i funcional.

El meu propòsit és fer que l'aplicació sigui tot el correcte possible per ser utilitzada pels alumnes de la Universitat. A part es tractarà de documentar correctament el projecte per si en un futur es vol ampliar de la mà dels professors o dels alumnes.

1.3 Estructura de la documentació

A continuació exposaré tots els apartats del que està composta aquesta memòria, amb un breu resum de què es veurà en cadascú d'ells.

1. Introducció, motivacions, propòsit i objectius del projecte: Aquest apartat ens situa en el marc del projecte. Ens explica les raons que justifiquen el desenvolupament del projecte i que se n'esperava obtenir.
2. Estudi de viabilitat: Es justifiquen els paràmetres que fan possible el desenvolupament del projecte, descriure els recursos necessaris, pressupostos inicials, viabilitat tecnològica i/o econòmica, els recursos humans, etc.
3. Metodologia: S'exposa la metodologia emprada pel desenvolupament del projecte.
4. Planificació: Aquesta etapa defineix l'estratègia seguida per arribar als objectius plantejats.
5. Marc de treball i conceptes previs: En aquest apartat donarem a conèixer una sèrie de conceptes necessaris per fer un bon seguiment del projecte i es descriuran els diversos aspectes relacionats amb el desenvolupament general del projecte que ajudaran a entendre millor els capítols següents.
6. Requisits del sistema: Es descriuran els requisits que ha de complir el sistema, tant funcionals com no funcionals. Ajudarà a l'usuari a saber tot allò que cal fer perquè el projecte compleixi tots i cadascun dels seus objectius.
7. Estudi i decisions: En aquest apartat s'ha de descriure el maquinari, les llibreries i/o el programari utilitzats durant el desenvolupament del projecte, i s'han de justificar les eleccions.
8. Anàlisi i disseny del sistema: Aquesta anàlisi estudiarà de manera detallada les necessitats del sistema, mentre que el disseny proposa una solució per al sistema.
9. Implementació i proves: En aquest apartat es detallen els problemes i les solucions apareguts en la implementació, així com, si és el cas, el model de classes i els algorismes més rellevants.

10. Implantació i resultats: Aquest apartat descriu amb detall el procés de desenvolupament que ha calgut dur a terme per implantar el sistema desenvolupat. L'apartat de resultats mostra clarament el gra d'assoliment dels objectius.
11. Conclusions: Diferenciar entre els objectius assolits i els requeriments inicials. S'expliquen les desviacions de la planificació original i es comenten els motius.
12. Treball futur: Comentar les possibles ampliacions, millores o treball futur del projecte.
13. Bibliografia: Referències utilitzades per desenvolupar el projecte.
14. Annexos.
15. Manual d'ús: En aquest apartat s'explicaran els requisits tècnics que ha de conèixer l'usuari i el funcionament del producte.

2. Estudi de viabilitat

Aquest projecte estava pensat perquè una sola persona el pogués desenvolupar. En aquest cas per jo mateix, l'autor. A conseqüència d'això, els costos afegits de tercers són nuls i només ha importat el meu temps lliure per dedicar-li al projecte.

No menysprear també el temps dedicat dels meus dos tutors que m'han aconsellat i orientat durant tot el desenvolupament del projecte.

Respecte a aspectes més tècnics he fet ús de programari lliure, amb llicències d'estudiant o totalment gratuïts, per tant sense costos.

A més com serà una aplicació docent de la Universitat de Girona estarà sobre una llicència lliure i oberta a tothom de manera gratuïta.

Resumint és un projecte viable i no lucratiu que només necessita hores de feina. Per tant l'estudi de viabilitat no és necessari en aquests cas.

3. Metodologia

Una metodologia refereix a un conjunt de conceptes, pràctiques i criteris que serveixen per enfocar un tipus de problema i que ens ajuda a enfrontar i resoldre problemes similars.

En el desenvolupament de software pot ser una estructura conceptual i tecnològica de suport definit. De les quals hi ha de molts tipus amb els seus avantatges i desavantatges.

En el meu cas he optat per una variant de la metodologia en cascada, amb alguna part iterativa.

La meva decisió per a aquesta metodologia a estat degut al funcionament de la Màquina Senzilla, ja que és una aplicació que perquè els usuaris puguin començar a interactuar i fer-la servir necessiten un nucli i una interfície operativa.

Fraccionar en parts i presentar versions precàries de la Màquina Senzilla, no aportava cap benefici a l'usuari final, ja que no podia presenciar tot el funcionament de la màquina.

Per això s'han descartat altres metodologies més iteratives.

La guia general de comportament d'aquesta metodologia és la següent:

1. Anàlisis dels requisits: En aquesta fase inicial, s'analitzen les necessitats dels usuaris finals per determinar quins són els objectius a cobrir.
2. Disseny del programa: S'estudia l'arquitectura del programa i també és la fase on s'estudien les eines que es faran servir a l'etapa de codificació.
3. Codificació: És la fase on implementem el codi font.
4. Proves: Fase de proves per detectar qualsevol error produït en l'anterior fase i corregir-lo.
5. Implementació del programa: És la fase on els usuaris finals executen l'aplicació.

6. Manteniment: Verificar si l'aplicació compleix amb les expectatives dels usuaris finals.

Com a variant del model tradicional en cascada s'ha fet un procés iteratiu entre les proves i la codificació amb els tutors i professors docents de l'assignatura en la qual s'imparteix la matèria de la Màquina Senzilla.

Un esquema d'aquesta metodologia seria:

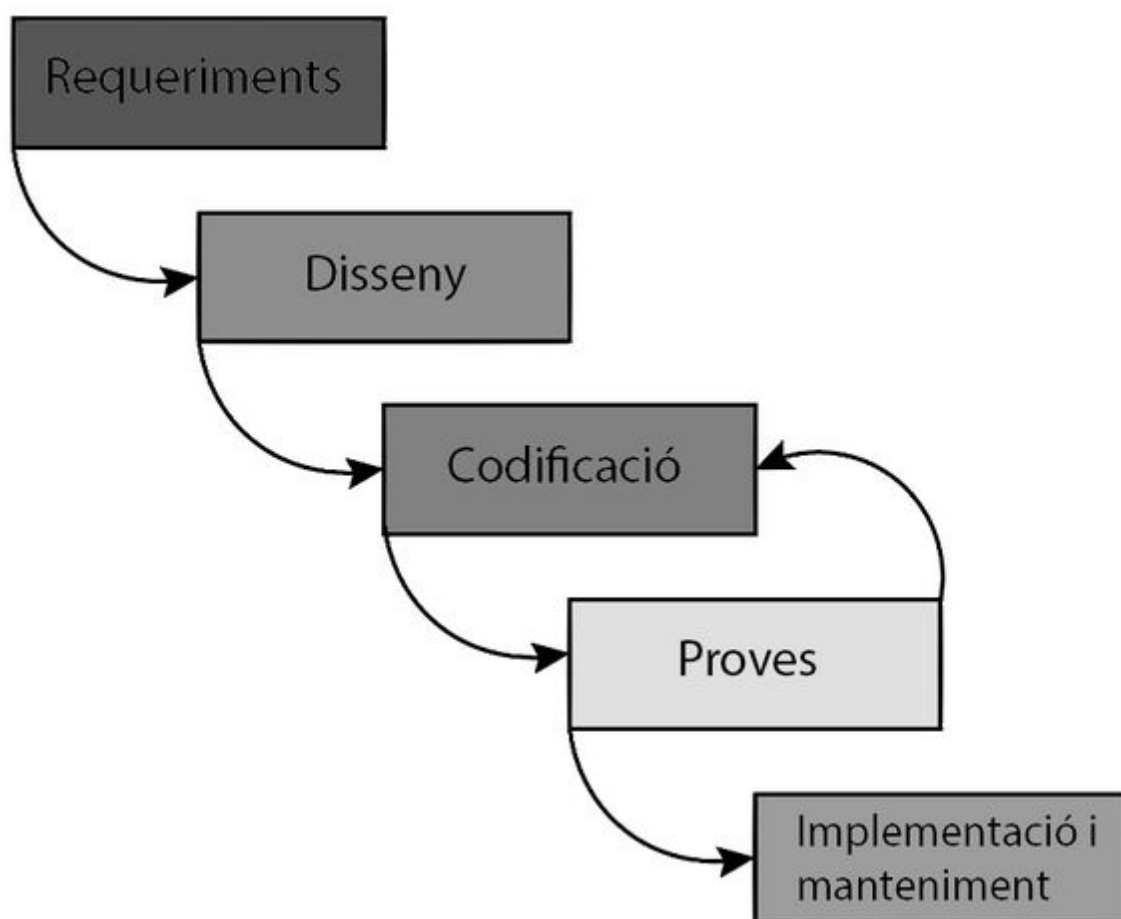


Figura 1: Esquema de la metodologia emprada al projecte.

4. Planificació

Aquest projecte va sorgir cap al març del 2016, on vaig començar a parlar amb els tutors de l'àrea d'arquitectura de computadors. En un inici estava més interessat en projectes d'aquest tipus però va sorgir el tema del Simulador de la Màquina Senzilla.

No tenia res a veure amb el que volia fer però la programació web i tot el que comporta en la part de servidor també em va agradar i vaig decidir-me.

Així doncs es va tirar el projecte cap endavant amb data prevista d'entrega al setembre del 2016.

En els primers mesos vam fer l'anàlisi dels requisits revisant l'antiga aplicació de la Màquina Senzilla i van sortir els requeriments i funcionalitats que hauria de tenir la nova aplicació.

Després en la fase de disseny de l'aplicació vam decidir quina tecnologia i quines eines faríem servir. Un cop sabut això, es va crear el diagrama de classes inicial i el model d'infraestructura que tindria.

La fase d'implementació va començar abans de l'estiu i per poc temps lliure i complexitat a l'hora de programar, es va decidir finalment posposar la presentació del projecte al febrer del 2017.

Fins al novembre del 2017, es va implementar tot el codi font de la Màquina Senzilla mentre els tutors opinaven i es canviaven coses sobre la marxa. Estàvem en una fase iterativa de codificació i proves.

A partir de principis de novembre l'aplicació va entrar en una fase alfa de proves per a professors docents de l'assignatura. A part d'estar oberta a Internet. Tot això per acabar d'arreglar problemes de l'aplicació.

Finalment la fase d'implementació per als usuaris finals (els alumnes) sortirà un cop el treball estigui finalitzat i entregat a mitjans de gener 2017.

El manteniment de l'aplicació es continuarà per la meua part si es tracta d'algun error molt greu del sistema. Altrament els professors i usuaris tindran accés al codi font i estaran lliures de modificar-lo.

Per il·lustrar d'una manera gràfica aquest procés he realitzat un diagrama de Gantt:

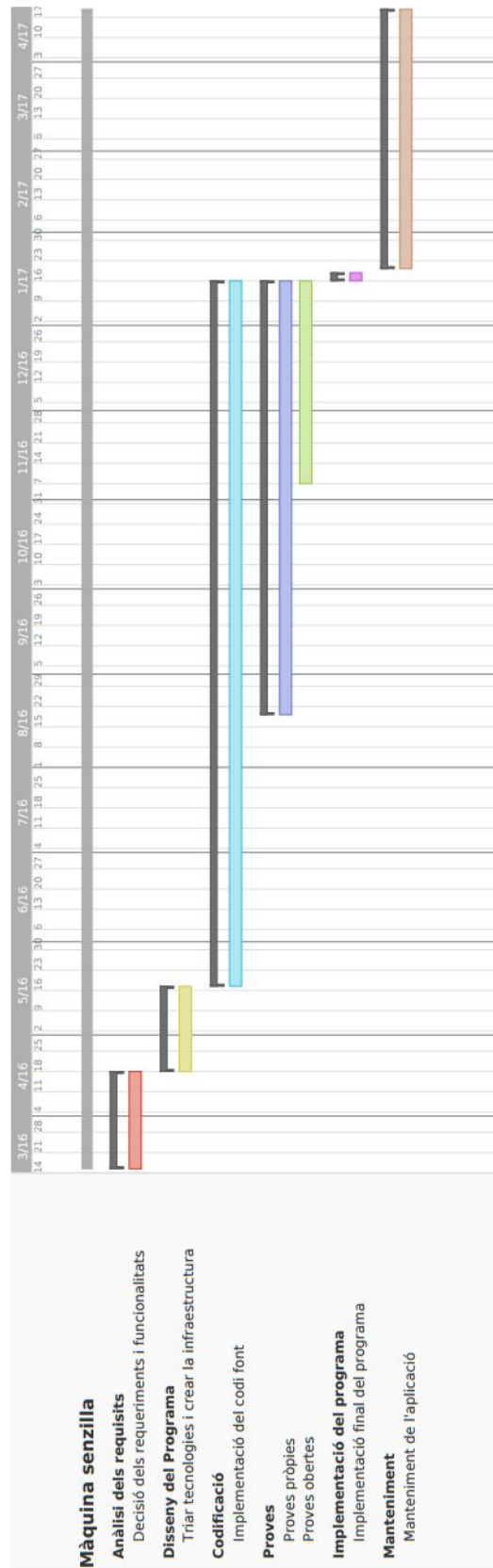


Figura 2: Diagrama de Gantt del temps invertit en el projecte.

5. Marc de treball i conceptes previs

En aquest punt explicarem els conceptes necessaris per a entendre el funcionament de totes les parts que envolten aquest projecte.

5.1 Arquitectura client-servidor

L'arquitectura client-servidor és un model d'aplicació distribuïda en el que les tasques es reparteixen entre els proveïdors de recursos o serveis, anomenats servidors, i els que ho demanen, anomenats clients.

Un client realitza peticions a un altre programa, el servidor, qui li dona resposta.

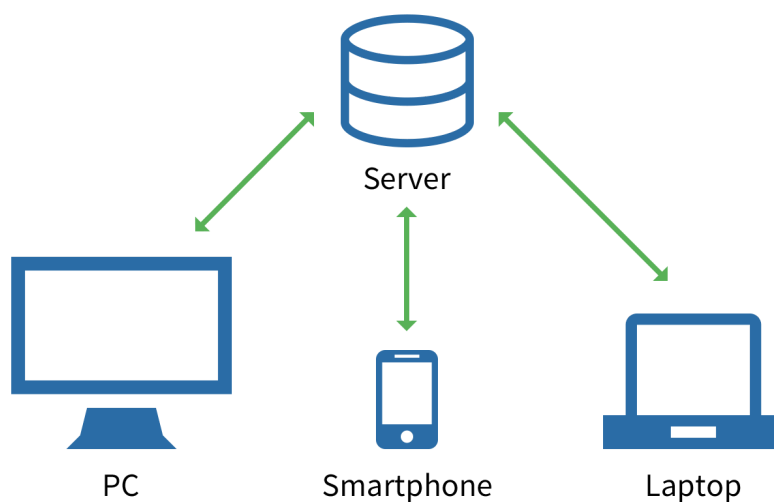


Figura 3: Concepte del funcionament entre clients i servidors.

Hi ha molts tipus de servidors, ja siguin de correu, d'intercanvi d'arxius, telefonia ip... Pel nostre cas particular utilitzarem un servidor web, que oferirà el seu servei mitjançant el protocol http, el qual estarà a l'espera per servir l'aplicació als clients que la sol·licitin.

Normalment en aquests tipus d'arquitectura la capacitat de procés està repartida entre els clients i el servidor.

En canvi, el disseny de la Màquina Senzilla que hem realitzat només treballa en la part de client, ja que un cop descarregada la pàgina del servidor, el client no li envia més peticions que processar. Fent que els servidors on està hostejada no gastin més recursos del compte.

Un resum de les característiques d'un servidor son:

- Passiu (esclau).
- Espera de peticions.
- Subministrament de serveis quan rep una petició.

Un resum de la part d'un client:

- Actiu (mestre).
- Envia peticions.
- En enviar una petició espera una resposta i la processa.

5.2 Serveis Web

Existeixen moltes definicions sobre el que són els serveis web, això demostra la seva complexitat a l'hora de donar una definició que encercli tot el que impliquen. Una possible definició seria parlar d'ells com un conjunt d'aplicacions o de tecnologies amb la capacitat d'interactuar amb la web. Aquestes aplicacions o tecnologies intercanvien dades entre elles amb l'objectiu d'oferir uns serveis. Els proveïdors ofereixen els seus serveis com procediments remots i els usuaris sol·liciten el seu servei cridant a aquests procediments a través de la Web.

Aquests serveis proporcionen mecanismes de comunicació estàndard entre diferents aplicacions, que interactuen entre si per a presentar informació dinàmica l'usuari.

Per proporcionar interoperabilitat i extensibilitat entre aquestes aplicacions, i que al mateix temps sigui possible la combinació per a realitzar operacions complexes, és necessària una arquitectura de referència estàndard.

Com hem comentat en l'apartat anterior, el servei web que oferim en aquest projecte és la mateixa aplicació del Simulador de la Màquina Senzilla, un servei docent enfocat a l'aprenentatge.

5.3 Single Page Application (SPA)

Un “Single Page Application” o aplicació de pàgina única és un lloc web composta d’una sola pàgina amb el propòsit de donar als usuaris una experiència més fluida, semblant a una aplicació d’escriptori.

En una aplicació d’aquest tipus, tots els codis HTML, Javascript i CSS es carreguen només una vegada. La pàgina no ha de tornar a recarregar un altre cop en cap punt del procés de la pàgina. També es pot donar el cas de què els recursos s’anessin carregant dinàmicament i s’afegissin a mesura que l’usuari navega per la web.

Per aconseguir això en aquest projecte hem utilitzat frameworks de Javascript que ho permeten. Que vénen a ser infraestructures que s’utilitzen en la part del client per evitar demanar dades al servidor o recarregar la pàgina per visualitzar els canvis. Més endavant parlarem del framework Backbone i el seu model MVC que ens permet fer tot això.

5.4 Patró Model, View, Controller (MVC)

És un patró d’arquitectura de software que separa les dades de la lògica de negoci d’una aplicació de la interfície d’usuari i el mòdul encarregat de gestionar els esdeveniments i les comunicacions.

Per això el MVC proposa la construcció de tres components diferents que són el model, la vista i el controlador, és a dir, per un costat defineix components per a la representació de la informació, i per un altre la interacció de l’usuari.

Aquest patró d’arquitectura del software es basa en les idees de reutilització de codi i la separació de conceptes, característiques que busquen facilitar la tasca en el desenvolupament d’aplicacions i el seu posterior manteniment.

De manera genèrica els components de MVC es podrien definir com:

- *El model*: la representació de la informació amb la qual el sistema opera, per tant gestiona tots els accessos a aquesta informació.

Envia a la vista la part d'informació que en cada moment es sol·licita per ser mostrada. Les peticions d'accés o manipulació d'informació arriben al model a través del controlador.

- *El controlador*: respon a esdeveniments (usualment accions de l'usuari) i invoca peticions al model quan es fa alguna sol·licitud sobre informació.

També pot enviar comandes a la seva 'vista' associada si es sol·licita un canvi en la forma en què es representa el model.

- *La vista*: presenta 'el model' (informació i lògica de negoci) en un format adequat per interactuar (usualment la interfície d'usuari) per tant requereix que dit 'model' la informació que deu presentar com a sortida.

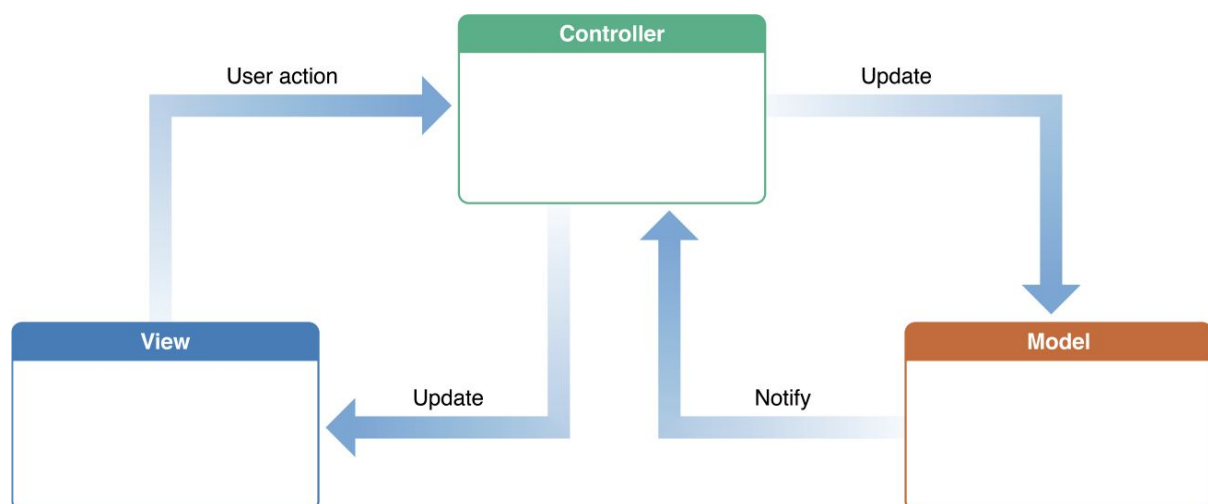


Figura 4: Interaccions entre el model la vista i el controlador del patró MVC.

Encare que originalment el patró MVC va ser desenvolupat per a aplicacions d'escriptori, ha estat àmpliament implementat i adaptat com a arquitectura per dissenyar i aplicacions web en els principals llenguatges de programació. S'han desenvolupat multitud de frameworks, comercials i no comercials, que implementen aquest patró.

Els primers frameworks MVC per a desenvolupament web plantejaven una visió de client lleuger en el qual, tant la vista, el model i el controlador funcionaven més en el servidor.

Com les tecnologies han madurat, ara existeixen frameworks més moderns com JavascriptMVC, Backbone o jQuery que permeten executar els components de MVC de forma parcial o totalment en la part del client.

5.5 Sistema de Control de Versions

Un sistema de control de versions és la gestió dels diversos canvis que es realitzen sobre els elements d'algun producte o d'una configuració d'aquest. Una versió, revisió o edició d'un producte que es troba en un moment del seu estat de desenvolupament o de modificació.

En el nostre àmbit informàtic de software, es refereix principalment a controlar les diferents versions del codi font.

Per a aquest projecte s'ha utilitzat un sistema de versions Git amb repositori privat a Bitbucket.com. Que vol dir que el projecte està emmagatzemat al "núvol" i té un històric de versions més antigues.

Els beneficis d'utilitzar un control de versions són:

- Tenir un mecanisme d'emmagatzematge de tot el projecte.
- Possibilitat de realitzar canvis sobre els elements emmagatzemats.
- Tenir un registre històric de les accions realitzades per a cada element o conjunt d'ells.

A part que, en un futur, si es continua en el desenvolupament d'aquest projecte, es podrà crear branques alternatives de l'aplicació i evolucionar-la.

6. Requisits del sistema

En aquest punt descriurem més extensament els requisits que ha de complir el sistema, tant requisits funcionals sobre el que fa directament l'aplicació com a requisits no funcionals.

Amb això aclarirem als usuaris que facin servir l'aplicació el seu ús i límits.

6.1 Requisits funcionals

En aquesta aplicació web, només tenim classificat un tipus d'usuari, ja que no es fa distinció entre professors i alumnes.

Per això representarem les funcionalitats segons el tipus d'interacció amb el sistema, si és d'entrada o de sortida.

Segons l'entrada d'informació per part de l'usuari l'aplicació haurà de:

- Editar la memòria RAM del sistema.
- Carregar un fitxer de text a la memòria RAM del sistema.
- Canviar els colors representatius dels senyals de control i el bus de dades.
- Canviar el mode de visió “normal” a “assemblador” de la màquina.
- Avançar els clock's en el funcionament de la màquina.
- Retrocedir els clock's en el funcionament de la màquina.
- Reiniciar l'estat de la màquina al estat inicial.
- Resetejar tota la màquina eliminant tota la informació.

Pel que és a la sortida:

- El sistema representarà en tot moment el flux de les dades. Representat per les instruccions i dades d'entrada.
- Informarà l'usuari de les transicions entre les diferents etapes del sistema, per mitjà d'una consola.
- Es mostraran els canvis en la RAM en temps real.
- El sistema avisarà a l'usuari d'errors d'entrada del fitxer i edició de la memòria, mitjançant un sistema de control.

6.2 Requisits no funcionals

Per a aquests requisits s'ha volgut remarcar alguns aspectes com:

- Resposta del temps, s'ha intentat fer l'aplicació el més optimitzada possible sense fer bucles dobles i evitant algorismes complexos.
- La capacitat d'emmagatzematge de l'aplicació, s'ha fet una aplicació lleugera per ser carregada ràpidament pels clients. S'ha evitat utilitzar imatges. S'han comprimit llibreries.
- Respecte a accessibilitat entre dispositius, s'ha fet primer pensant en resolucions d'escriptori, però està adaptat fins a *tablets*. A partir de resolucions més baixes de 1024x768 no es garanteix una correcta visualització.

A més per aquest mateix motiu, en la visualització de la màquina s'ha evitat fer ús de funcions només compatibles amb entorns d'escriptori i s'han mostrat les dades de forma més accessible per a tots els dispositius.

- Disseny, s'ha utilitzat llibreries que donen un disseny molt més modern a l'aplicació.

- Compatibilitat amb els navegadors, l'aplicació és compatible amb tots els navegadors moderns, menys amb Internet Explorer a causa d'algunes incompatibilitats amb les imatges SVG i les seves animacions.

7. Estudis i decisions

En aquest apartat descriurem les tecnologies utilitzades per fer aquest projecte i el perquè d'aquestes.

7.1 Llenguatge de programació

Per a l'estructura de l'aplicació web no vaig tenir dubtes en utilitzar HTML i CSS pel disseny, en canvi, per a la part del nucli de l'aplicació vaig estar dubtant entre utilitzar PHP o Javascript, ja que eren els llenguatges web que havia vist durant la carrera.

PHP és un llenguatge amb una corba d'aprenentatge baixa, és molt pràctic i ràpid però només interpretat a la part del servidor, això pot fer que el servidor es col·lapsi quan el nombre de peticions de la pàgina augmenta.

En canvi, Javascript, que no segueix el model de programació “tradicional” i és més lent d'aprendre (comparat amb PHP) s'executa en la part del client, solucionant els problemes que podria tenir el servidor.

En el nostre cas, l'aplicació de la màquina senzilla no ha de guardar res a bases de dades al servidor ni utilitzar altres serveis que aquest ofereixi. Per tant utilitzar PHP no és necessari.

Per això l'elecció que he fet per desenvolupar el nucli de l'aplicació ha estat Javascript.

7.2 Frameworks

Un cop seleccionat el llenguatge principal de programació Javascript vaig estar buscant frameworks que ajudessin a estructurar el projecte i orientats en el model de MVC i van aparèixer l'AngularJS, Backbone i Ember entre altres.



Figura 5: Logotips en ordre d'esquerra a dreta, AngularJS, Backbone i Ember.

En aquestes tres opcions vaig començar a desenvolupar en AngularJS, ja que és el més potent dels tres i el que més popularitat té actualment. Vaig continuar però la corba d'aprenentatge és alta i injecció directa en l'HTML no em va convèncer, preferia separar el nucli de l'aplicació del DOM.

Sobre Ember vaig documentar-me i vaig veure que era un llenguatge molt rígid i també poc documentat.

En canvi ja vaig treballar amb Backbone durant la carrera i és un framework molt lleuger i molt intuïtiu d'aprendre i aplicar.

Tornant a repassar les meves opcions i pel tipus de projecte, no vaig veure la necessitat de fer servir un framework tan potent com l'AngularJS, per això al final es va escollir Backbone.

7.3 Framework front-end

Els frameworks front-end serveixen per al disseny dels llocs i pàgines webs. Contenen plantilles de disseny amb tipografia, formularis, botons, requadres de navegació i altres elements de disseny basats en HTML i CSS així com extensions de Javascript opcionals.

En un principi es va pensar a utilitzar el framework UIKit però finalment es va fer servir Bootstrap per a l'extensa documentació que existeix d'ell.

7.4 Llibreries

En aquest apartat explicarem totes les llibreries extres que hem utilitzat en el projecte.

- jQuery: permet simplificar la manera d'interactuar amb els documents HTML, manipular el DOM, controlar esdeveniments, desenvolupar animacions i afegir interacció amb la tècnica AJAX a les pàgines web.

És un requisit si s'utilitza el framework Bootstrap i Backbone.

- underscore: una dependència de Backbone, conté funcions extres que ens ajuden en la programació.
- Requirejs: és una eina que ens permet aïllar mitjançant mòduls els components de la nostra aplicació client i resoldre aquestes dependències d'aquests mateixos.

El que ens permet una càrrega dinàmica de les dependències, la seva descàrrega automàtica i el maneig de "timeouts" i càrrega de "fallbacks".

- jsColor: és un selector de colors molt simple amb suport "touch" de dispositius tablets i mòbils.

7.5 Imatges Vectorials SVG

Per a aquest projecte s'ha escollit el format d'imatge SVG (Scalable Vector Graphics) que són una família d'especificacions d'un format de fitxer basat en el XML per descriure gràfics vectorials bidimensionals, tan estàtics com dinàmics.

Les imatges SVG i els seus comportaments es defineixen en arxius de text XML. Això significa que dins aquests es pot cercar, indexar, xifrar o fins i tot comprimir.

Gràcies a això el pes de l'aplicació de la màquina senzilla és mínim, ja que les imatges són textos XML que van canviant de propietats segons canvia l'estat de la màquina.

Excepte Internet Explorer, que dona algunes incompatibilitats amb SVG sobretot en les animacions, la resta de navegadors més moderns suporten les imatges SVG.

Un exemple aplicat dins del HTML seria:

```
<!DOCTYPE html>
<html>
<body>

<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%"
        style="stop-color:rgb(255,255,0);stop-opacity:1" />
      <stop offset="100%"
        style="stop-color:rgb(255,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="45" font-family="Verdana"
    x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```



Figura 6: Exemple de codi i de representació d'un SVG.

7.6 Entorn de desenvolupament

Per desenvolupar el projecte vaig escollir el IDE WebStorm, un dels entorns de desenvolupament de Javascript més potents del mercat. En un principi és un software de pagament però amb la llicència d'estudiant et donen un any de període gratuït.

Després es va crear un repositori Git privat a Bitbucket, per mantenir un control de versions i que el projecte estigues ben emmagatzemat al “núvol”.

Gràcies al fet que WebStorm incorpora la tecnologia Git podia pujar i actualitzar el projecte des del mateix programa.

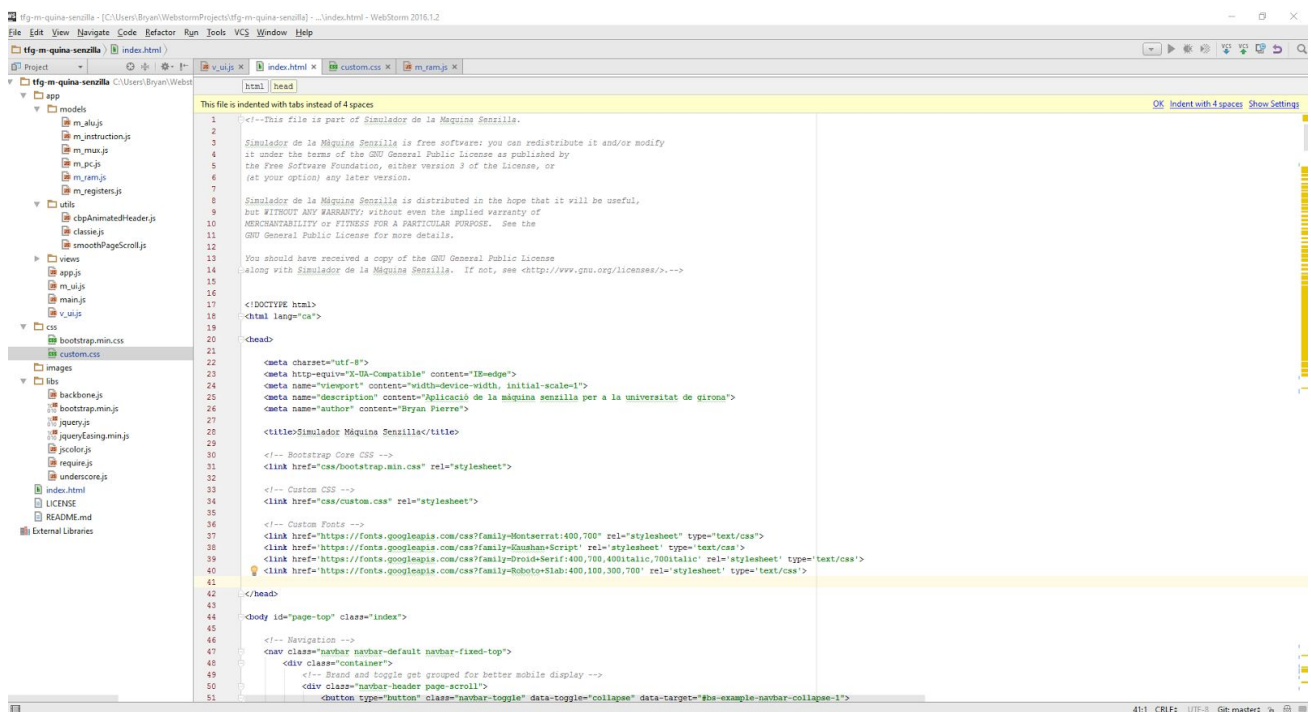


Figura 7: IDE WebStorm.

The screenshot shows the Bitbucket web interface for a private repository named 'TFG Màquina Senzilla'. The left sidebar contains navigation links such as 'Clonar', 'Crear branch', 'Crear pull request', 'Comparar', 'Derivación', 'Resumen', 'Fuente', 'Commits' (selected), 'Ramas', 'Pull requests', 'Pipelines', 'Descargas', and 'Opciones'. The main content area displays the 'Commits' page for user 'Bryan Pierre'. A vertical timeline on the left of the commit list shows a sequence of green arrows pointing downwards, indicating the chronological order of commits. The commit list table includes columns for 'Autor', 'Commit', 'Mensaje', 'Fecha', and 'Builds'. The commits are listed in descending order of date, starting from 2017-01-08 at the top and ending with 2016-07-28 at the bottom. The messages are truncated, showing 'etc ...' for most entries, with a few providing more detail like 'vistas de instrucciones i pc fetes' and 'Moltes coses'.

Autor	Commit	Mensaje	Fecha	Builds
Bryan	4971827	etc ...	2017-01-08	
Bryan	699c763	etc ...	2016-12-19	
Bryan	a145425	etc ...	2016-12-06	
Bryan	73f18ef	etc ...	2016-11-27	
Bryan	b795d75	etc ...	2016-11-20	
Bryan	854d7b7	etc ...	2016-11-17	
Bryan	ea17fe8	etc ...	2016-11-15	
Bryan	e18ad4d	etc ...	2016-11-14	
Bryan	8b46468	etc ...	2016-11-12	
Bryan	3aa23b9	etc ...	2016-11-12	
Bryan	e198685	etc ...	2016-11-12	
Bryan	b16b9d	etc ...	2016-11-11	
Bryan	c75fb6f	etc ...	2016-11-05	
Bryan	0b4d487	etc ...	2016-11-05	
Bryan	1aeceac3	etc ...	2016-11-01	
Bryan	d2bd117	etc ...	2016-11-01	
Bryan	3a5284c	etc ...	2016-10-15	
Bryan	5e55482	etc ...	2016-10-12	
Bryan	f7d9ae4	etc ...	2016-09-30	
Bryan	9378537	etc ...	2016-09-17	
Bryan	a181802	vistas de instrucciones i pc fetes	2016-08-23	
Bryan	9e754fd	Moltes coses	2016-07-28	

Figura 8: Repositori privat a Bitbucket.com

7.7 Servidor

En aquest projecte s'ha utilitzat un servidor real obert a Internet. És un servidor d'Amazon Web Services. Són servidors de pagament per recursos però durant el primer any són gratuïts. De moment l'aplicació romandrà en aquest servidor fins que els professors hostegin la pàgina en els servidors de la universitat.

El servidor que hem configurat és una instància d'Ubuntu Server. Aquesta decisió és una decisió personal que bé donada pel temps que li he dedicat a aquest sistema i l'experiència que tinc en configurar servidors webs en aquesta distribució.

Per servidor web he escollit Apache2 que és OpenSource. Aquesta decisió bé per l'experiència prèvia respecte al software. Més endavant en l'apartat d'implantació es veurà com es fa tota la configuració del servidor d'Amazon i la del Apache.

Per a la nostra aplicació no necessitem res més, ni base de dades, ni servidor de correu, ni php, ni altres programes i/o aplicacions. Amb carregar els fitxers en el servidor web i comprovar que el port 80 (http) estigui obert els usuaris ja podran utilitzar l'aplicació.



Linux



Figura 9: Logotips del sistema operatiu Linux i del servidor Web Apache2.

8. Anàlisi i Disseny del sistema

En aquest apartat es farà l'anàlisi dels requisits funcionals del projecte. S'il·lustrarà amb diagrames de casos d'ús i les seves fitxes corresponents, a part de presentar el diagrama de classes.

8.1 Anàlisi

8.1.1 Diagrames de casos d'ús

En aquest projecte només apareix un tipus d'actor, l'usuari que utilitza l'aplicació. Per tant només tenim les opcions que pot executar ell, que són:

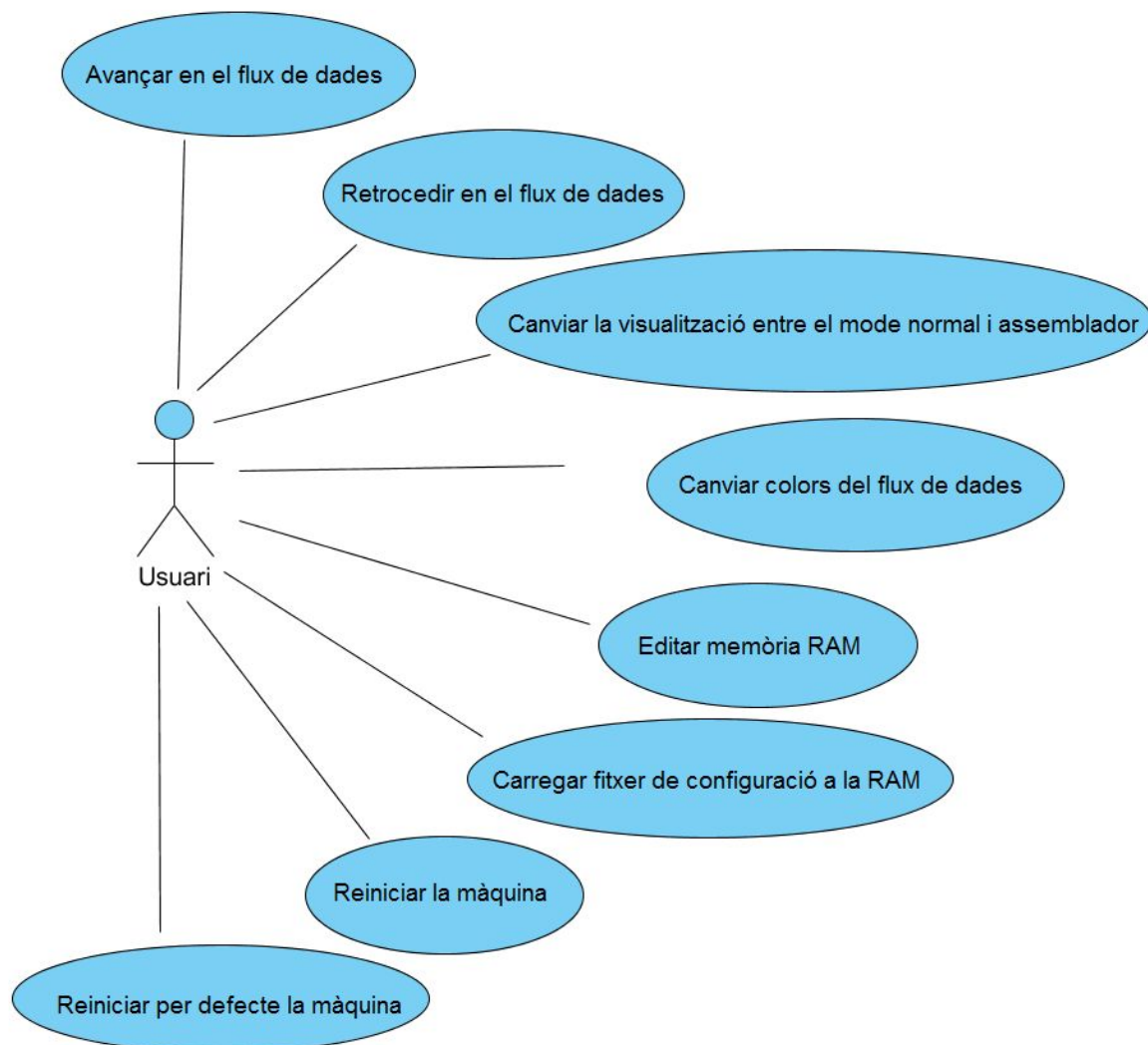


Figura 10: Diagrama de cas d'ús general.

Ara descriurem més detalladament les funcionalitats més rellevants del projecte de la màquina senzilla.

Avançar en el flux de dades

Per avançar en el flux de dades de forma efectiva és necessari haver carregat prèviament un fitxer amb èxit a la configuració de la RAM o bé editar i guardar aquesta configuració des de la mateixa web.

En cas contrari s'indicarà un error.

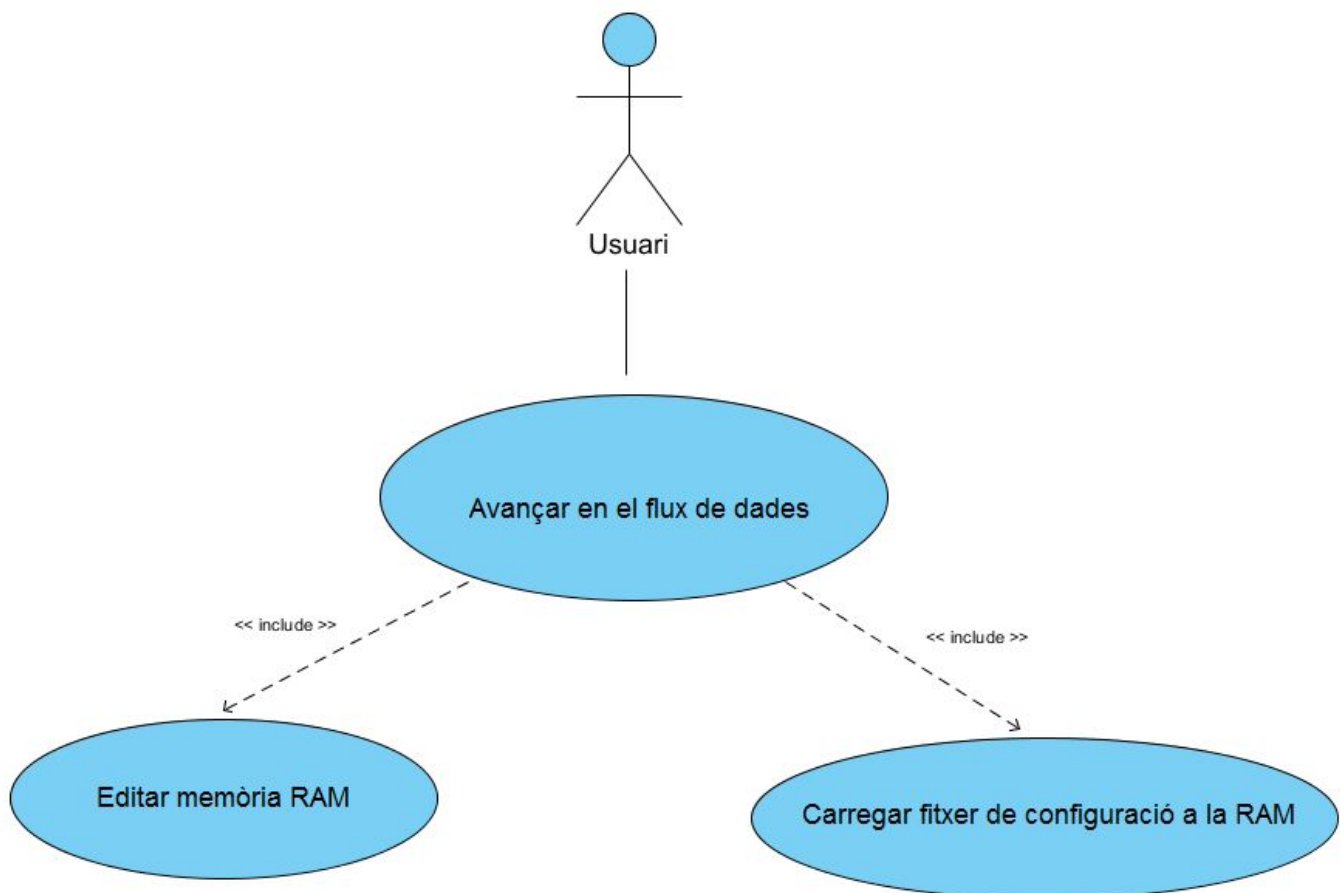


Figura 11: Diagrama de cas d'ús d'avançar en el flux de dades

Retrocedir en el flux de dades

Per igual manera que en el cas anterior, si volem retrocedir en els estats de la màquina, haurem de carregar una configuració prèvia i avançar uns quants passos en el flux de l'aplicació per poder retrocedir amb èxit.

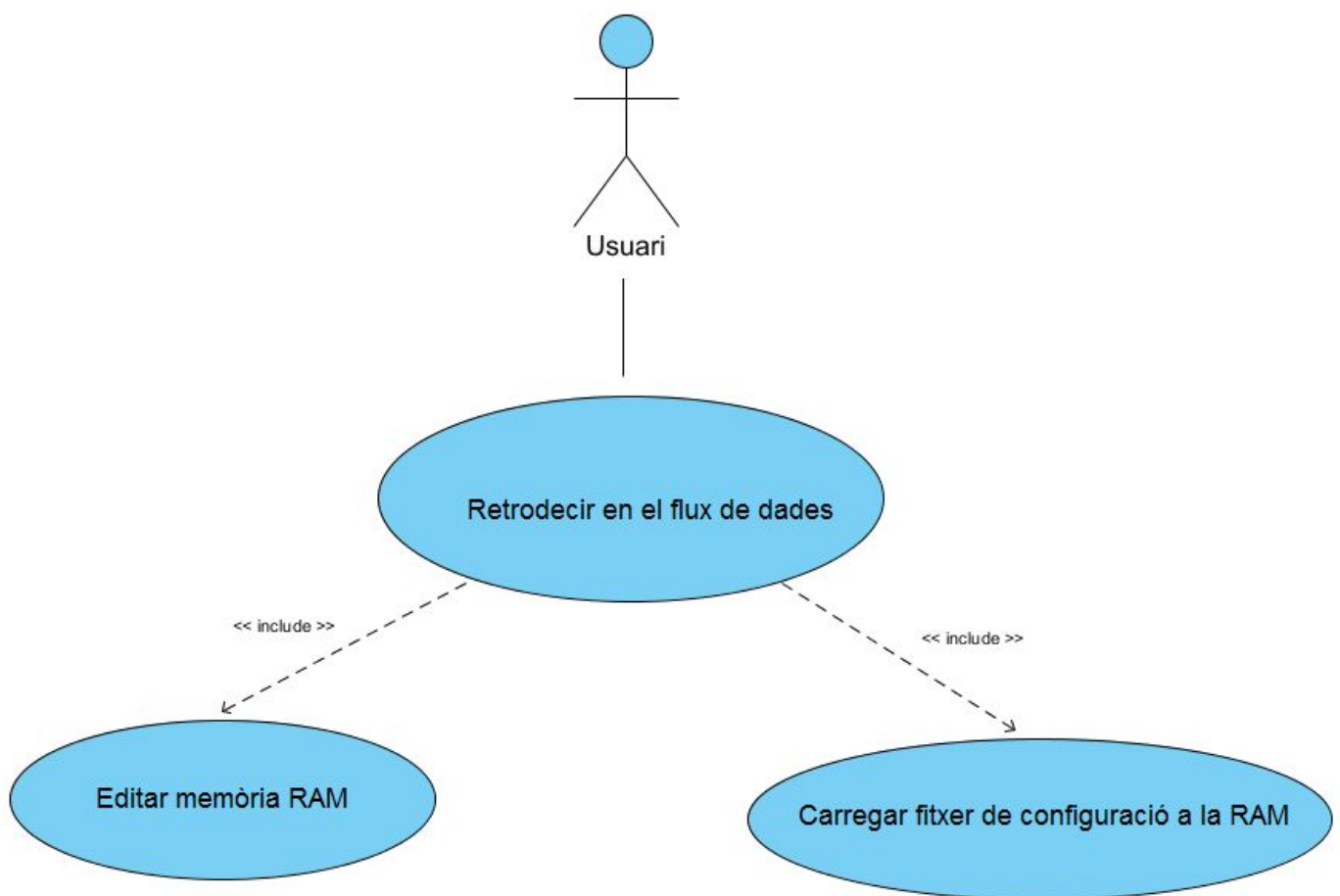


Figura 12: Diagrama de cas d'ús de retrocedir en el flux de dades.

Carregar fitxer de configuració a la RAM

Per a la càrrega del fitxer, es necessita un fitxer de tipus .txt amb una sintaxi específica, ja que el sistema executarà un compilador o “parser” d’aquest fitxer.

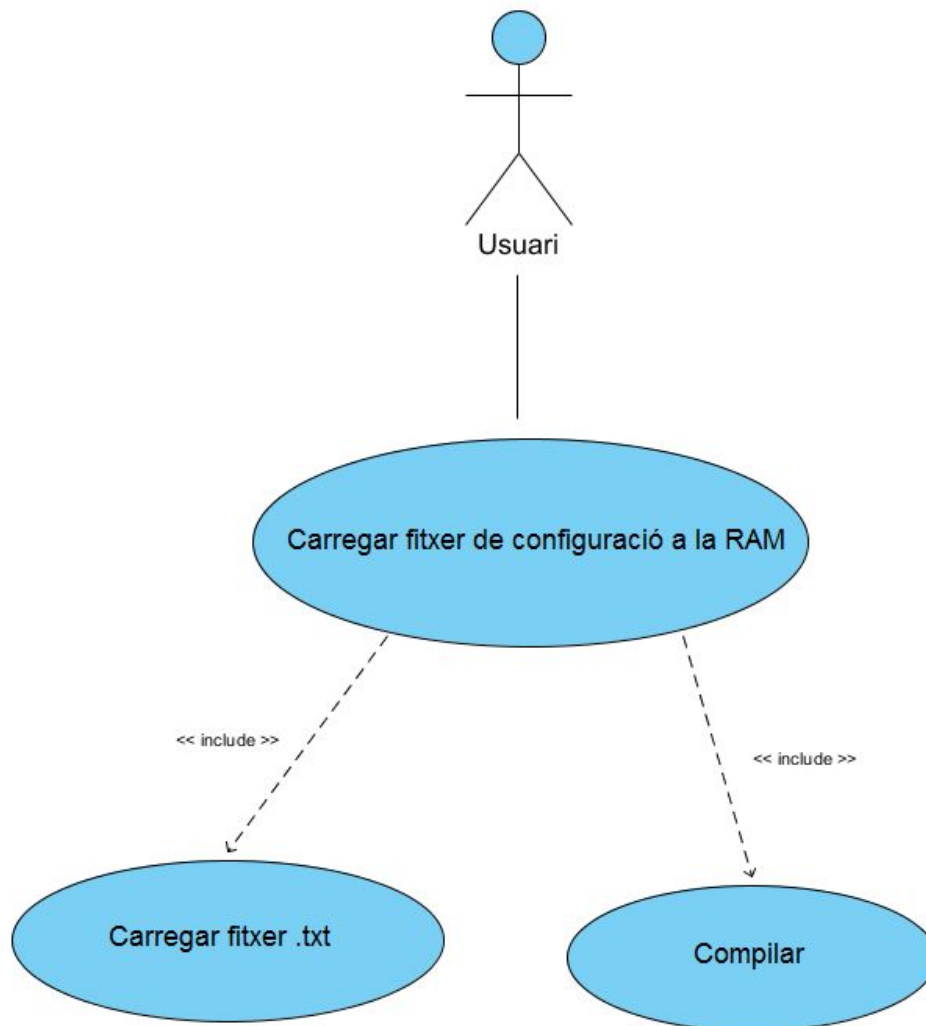


Figura 13: Diagrama de cas d'ús carregar fitxer de configuració RAM.

Editar memòria RAM

En l'edició de la memòria RAM ens estalviem la part de carregar el fitxer i processar-lo. Simplement l'usuari escriurà una configuració en l'apartat corresponent i en guardar les dades aquestes seran interpretades i processades pel compilador del sistema.

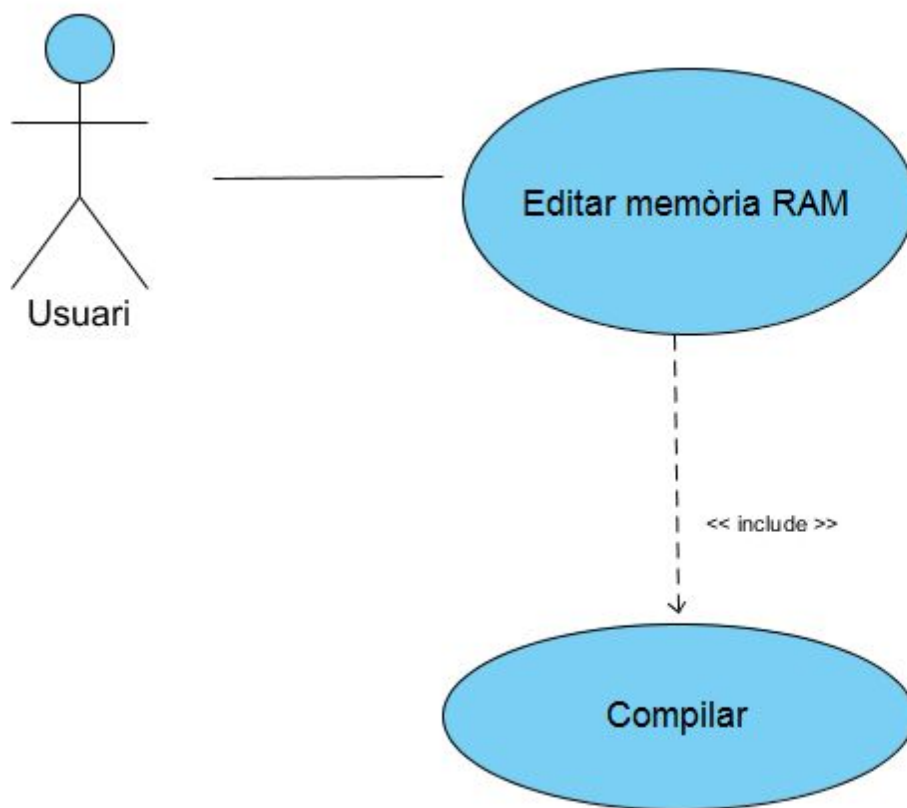


Figura 14: Diagrama de cas d'ús de editar memòria RAM.

Els altres casos d'ús no tenen molt més desenvolupament. En tot cas veurem el seu funcionament més detallat en les seves fitxes d'ús.

8.1.2 Fitxes de cas d'ús

A continuació mostrarem totes les fitxes dels casos d'ús de l'aplicació, on es detallaran els actors, una descripció de la funcionalitat, les precondicions que ha de complir, el flux principal d'esdeveniments i la postcondició final. Tot des d'un put de vista informàtic.

Avançar en el flux de dades

CAS D'US	Avançar en el flux de dades
Versió	Informàtic.
Descripció	L'usuari de l'aplicació podrà veure com avança el flux de dades a la màquina senzilla.
Actors	Usuari
Precondició	Per avançar amb èxit, l'usuari haurà d'haver carregat una configuració prèvia a la RAM, altrament donarà error.
Flux Principal	<ol style="list-style-type: none"> 1. Pulsar botó "clock +" per enviar l'ordre d'avançar en el flux 2. Si existeix o no una configuració prèvia a la RAM <ol style="list-style-type: none"> 2.1 Existeix <ol style="list-style-type: none"> 2.1.1 Si el següent cicle no surt del rang de les instruccions <ol style="list-style-type: none"> 2.1.1.1 No surt <ol style="list-style-type: none"> 2.1.1.1.1 Si sobreescriu una dada per sobre dels 16bits <ol style="list-style-type: none"> 2.1.1.1.1.1 No més de 16bits <ol style="list-style-type: none"> 2.1.1.1.1.1.1 Es mostra el següent cicle en el flux de dades segons les instruccions i variables carregades a la RAM 2.1.1.1.1.2 Més de 16bits <ol style="list-style-type: none"> 2.1.1.1.1.2.1 Error("No es pot executar, passa el limit de bits") 2.1.1.1.2 Surt <ol style="list-style-type: none"> 2.1.1.1.2.1 Error ("S'està executant fora del rang d'instruccions") 2.2 No existeix <ol style="list-style-type: none"> 2.2.1 Error ("Primer edita o afegeix un fitxer de configuració a la RAM")
Postcondició	L'usuari visualitza els canvis si s'efectuen a la màquina senzilla.

Retrocedir en el flux de dades

CAS D'US	Retrocedir en el flux de dades
Versió	Informàtic.
Descripció	L'usuari de l'aplicació podrà veure com retrocedeix el flux de dades a la màquina senzilla.
Actors	Usuari
Precondició	Per retrocedir amb èxit, l'usuari haurà d'haver carregat una configuració prèvia a la RAM i haver avançat almenys un cicle de temps, altrament donarà error.
Flux Principal	<ol style="list-style-type: none"> 1. Pulsar botó “clock -” per enviar l'ordre de retrocedir en el flux 2. Si existeix o no una configuració prèvia a la RAM <ol style="list-style-type: none"> 2.1 Existeix <ol style="list-style-type: none"> 2.1.1 Si el cicle actual no és zero <ol style="list-style-type: none"> 2.1.1.1 No és zero <ol style="list-style-type: none"> 2.1.1.1.1 Es mostra l'anterior cicle en el flux de dades segons les instruccions i variables carregades a la RAM 2.1.1.2 És zero <ol style="list-style-type: none"> 2.1.1.2.1 Error (“No pots anar més enrere”) 2.2 No existeix <ol style="list-style-type: none"> 2.2.1 Error (“Primer edita o afegeix un fitxer de configuració a la RAM”)
Postcondició	L'usuari visualitza els canvis si s'efectuen a la màquina senzilla.

Editar memòria RAM

CAS D'US	Editar memòria RAM
Versió	Informàtic.
Descripció	L'usuari de l'aplicació podrà editar i carregar una configuració a la memòria RAM
Actors	Usuari
Precondició	Per carregar correctament a la memòria RAM, s'haurà de fer amb la sintaxi correcta de l'aplicació, altrament donarà error.
Flux Principal	<ol style="list-style-type: none"> 1. Pulsar botó "Editar memòria" per enviar l'ordre d'obrir la pantalla d'edició. 2. Entrar les dades 3. Pulsar el botó "Guardar" 4. S'executa el compilador 5. Si la sintaxi és correcta <ol style="list-style-type: none"> 5.1 És correcta <ol style="list-style-type: none"> 5.1.1 Es carrega a la RAM la nova configuració 5.2 No és correcta <ol style="list-style-type: none"> 2.2.1 Error: Retorna els errors detectats
Postcondició	L'usuari edita i carrega correctament la nova configuració a la RAM

Carregar fitxer de configuració a la RAM

CAS D'US	Carregar fitxer de configuració a la RAM
Versió	Informàtic.
Descripció	L'usuari de l'aplicació podrà carregar un fitxer de configuració a la memòria RAM
Actors	Usuari
Precondició	Per carregar correctament a la memòria RAM, el fitxer haurà d'estar en format .txt i tenir la sintaxis correcta, altrament donarà error.
Flux Principal	<ol style="list-style-type: none"> 1. Pulsar botó “Seleccionar archivo” per enviar l'ordre d'obrir la pantalla de selecció de fitxers del nostre SO. 2. Seleccionar el fitxer 3. Si és format .txt <ol style="list-style-type: none"> 3.1 És format .txt <ol style="list-style-type: none"> 3.1.1 S'executa el compilador 3.1.2 Si la sintaxi és correcta <ol style="list-style-type: none"> 3.1.2.1 És correcta <ol style="list-style-type: none"> 3.1.2.1.1 Es carrega a la RAM la nova configuració 3.1.2.2 No és correcta <ol style="list-style-type: none"> 3.1.2.2.1 Error: Retorna els errors detectats 3.2 No és format .txt <ol style="list-style-type: none"> 3.2.1 Error (“Només s'accepten fitxers .txt”)
Postcondició	L'usuari carrega la nova configuració a la RAM des d'un fitxer tipus .txt

Reiniciar la màquina

CAS D'US	Reiniciar la màquina
Versió	Informàtic.
Descripció	L'usuari de l'aplicació podrà reiniciar l'estat visual de la màquina, a la primera etapa després de carregar les instruccions a la RAM, sense esborrar el contingut d'aquesta.
Actors	Usuari
Precondició	--
Flux Principal	<ol style="list-style-type: none"> 1. Pulsar botó "Reset" per enviar l'ordre de reiniciar la màquina. 2. Es reinicia visualment l'estat de la màquina i el PC apunta a zero.
Postcondició	L'usuari es troba en l'estat inicial de la configuració de la màquina senzilla, carregada a la RAM.

Reiniciar per defecte la màquina

CAS D'US	Reiniciar per defecte la màquina
Versió	Informàtic.
Descripció	L'usuari de l'aplicació podrà reiniciar l'estat visual de la màquina, eliminant així també el contingut de la RAM.
Actors	Usuari
Precondició	--
Flux Principal	<ol style="list-style-type: none"> 1. Pulsar botó "Kill" per enviar l'ordre de reiniciar per defecte la màquina. 2. Es reinicia visualment l'estat de la màquina, el PC apunta a zero i els valors de la RAM es posen per defecte.
Postcondició	L'usuari es troba en l'estat inicial de la configuració de la màquina senzilla.

Canviar mode de visualització normal a assemblador i viceversa

CAS D'US	Canviar mode de visualització normal a assemblador i viceversa
Versió	Informàtic.
Descripció	L'usuari de l'aplicació podrà canviar els modes de reproducció de la màquina senzilla.
Actors	Usuari
Precondició	--
Flux Principal	<ol style="list-style-type: none"> 1. Pulsar botó "Assemblador" per enviar l'ordre de canviar de mode. 2. Es mostra el canvi a la màquina senzilla.
Postcondició	S'ha canviat el mode de representació de la màquina senzilla

Canviar colors del flux de dades

CAS D'US	Canviar colors del flux de dades
Versió	Informàtic.
Descripció	L'usuari de l'aplicació podrà canviar els colors dels principals fluxos de la màquina senzilla.
Actors	Usuari
Precondició	--
Flux Principal	<ol style="list-style-type: none"> 1. Pulsar els botons de la llegenda 2. Seleccionar el color desitjat
Postcondició	En la següent iteració cap endavant o enrere de la màquina els colors es mostraran amb la nova configuració.

8.2 Disseny del sistema

8.2.1 Diagrama de classes

Per al disseny de la màquina senzilla hem estructurat el projecte segons els elements del que està composta la màquina. Tot el diagrama queda de la següent manera:

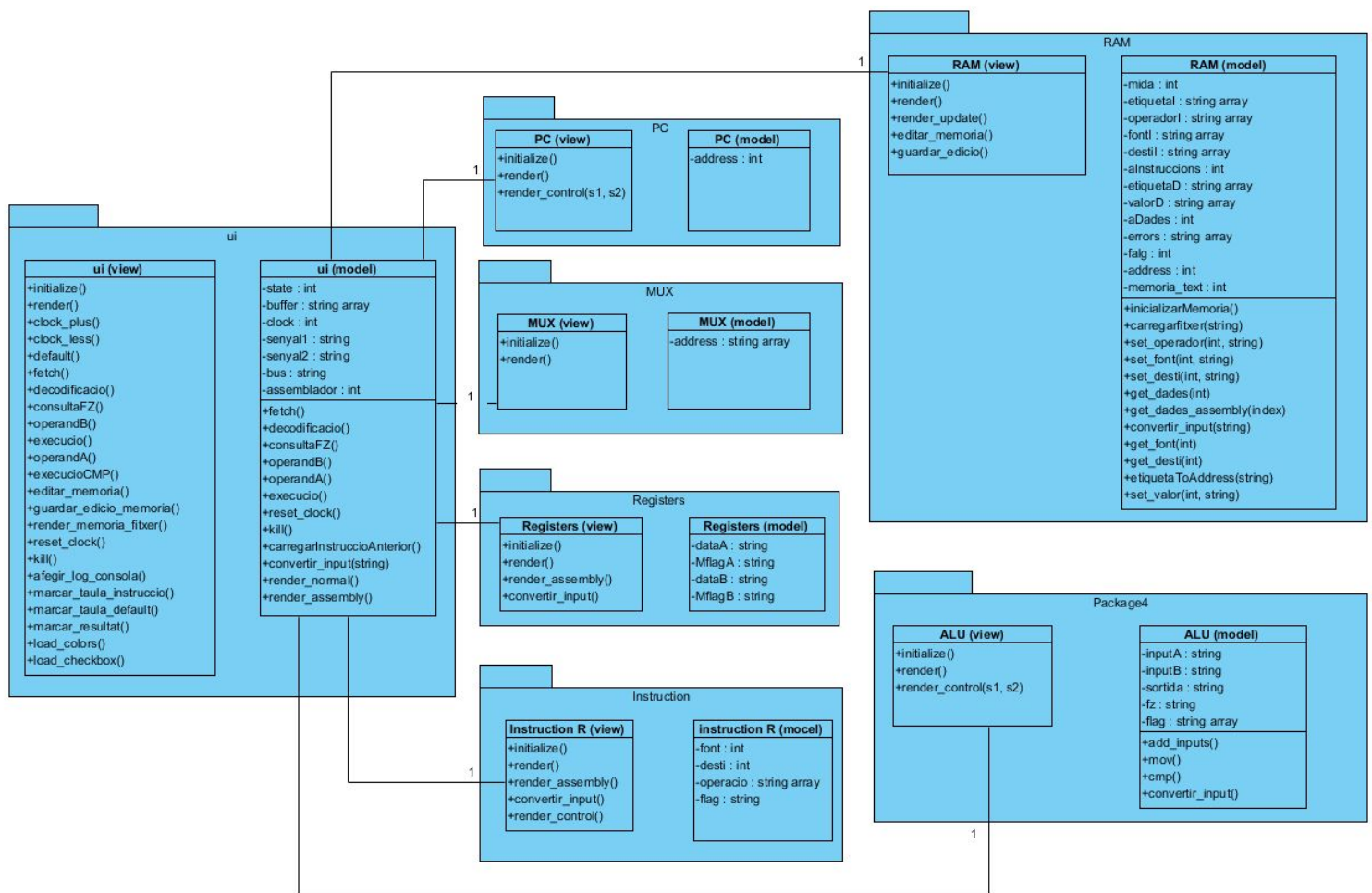


Figura 15: Diagrama de classes general del projecte de la màquina senzilla.

En aquesta imatge veiem que el disseny de classes està separat en blocs, cada bloc representa la vista i el model de cada element de la màquina senzilla, aquest és el disseny que proposa Backbone, on el controlador està integrat en la vista.

A continuació detallarem més cada bloc per veure els seus atributs i funcions primàries.

Per començar parlarem de la classe UI. Que és la classe que engloba tot el funcionament de la màquina senzilla.

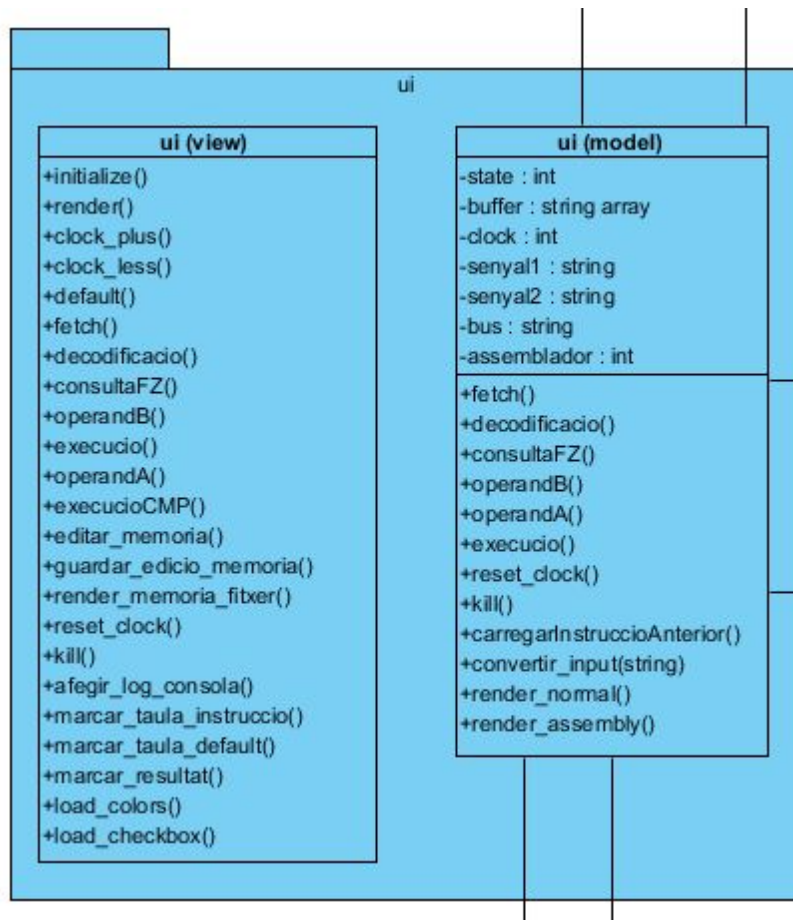


Figura 16: Part del diagrama de classes, específicament la classe Ui.

Aquesta classe està composta pels altres elements del sistema, els quals tenen una relació d'un a un.

A part, aquesta classe conté atributs globals de l'aplicació com pot ser el clock, l'estat en què es troba la màquina, el buffer on es guarden les instruccions anteriors i fins i tot variables per a la representació de la màquina com són les variables senyal1, senyal2, bus o assemblador.

Pels que són les funcions, l'UI executa segons quin estat de la màquina la funció corresponent.

La RAM és un altre element molt important de l'aplicació, ja que moltes accions de la màquina senzilla fan canvis a la RAM i la visualització d'aquesta està molt present en tot moment.

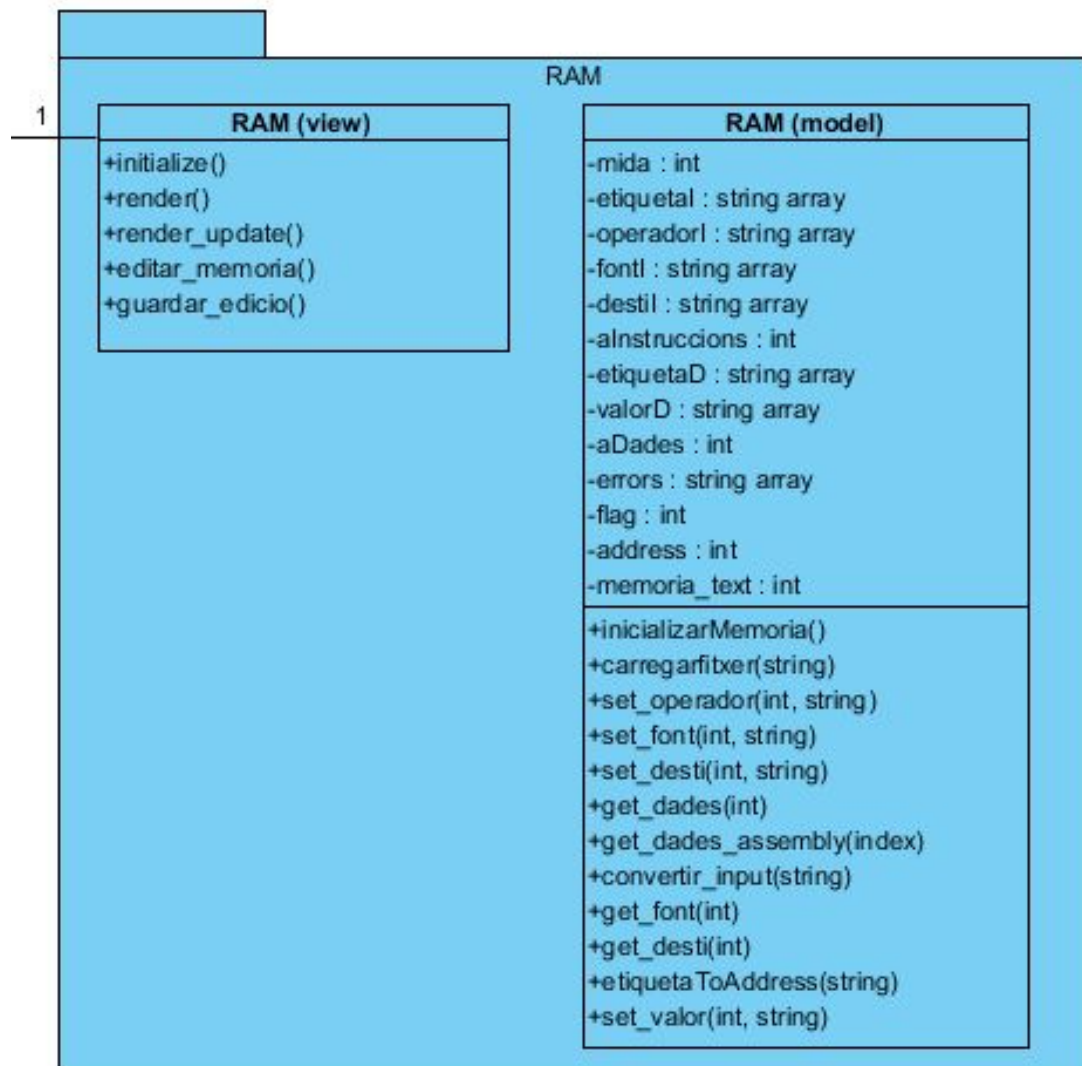


Figura 17: Part del diagrama de classes, específicament la classe RAM.

Podem veure que en la RAM es guarden en taules les instruccions i les variables, a més de variables globals com la mida total de la memòria o on començant les dades i acaben les instruccions.

Les funcions bàsiques de la RAM són els accessos per consultar dades i els accessos per escriure resultats.

Pel que són els altres elements de la màquina senzilla, són més bàsics, però aquesta separació afegeix un nivell més d'abstracció que fa que l'aplicació sigui més fàcil de continuar i entendre en un futur.

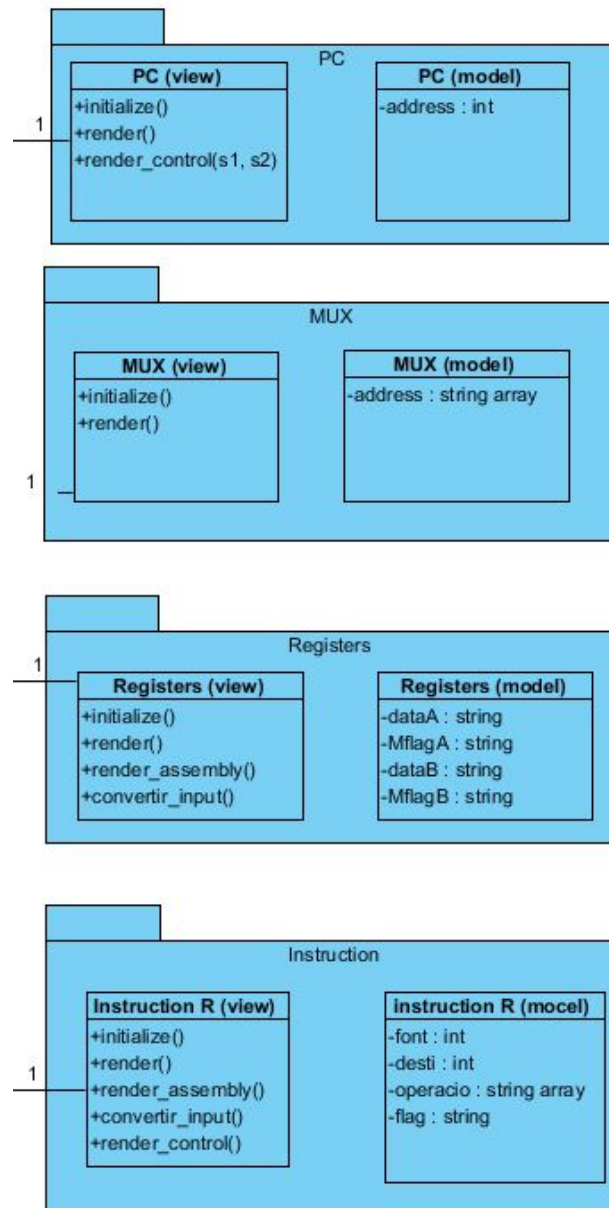


Figura 18: Les altres classes que componen el projecte.

Destacar que la manca d'operacions d'accés de consulta i d'escriptura, és pel fet que Backbone compta amb funcions que ens permeten fer-ho, alleugerin el codi i evitant funcions trivials.

9. Implementació i proves

Per a aquest apartat s'explicarà pas a pas la metodologia tècnica pel desenvolupament del projecte, els problemes sorgits i les seves respectives solucions aportades.

9.1 Estructura de directoris del projecte

Primer de tot abans de començar a programar i crear les classes hem de tenir una estructura de directoris sòlida i clara que faciliti i separi els diferents elements de l'aplicació. El resultat obtingut és el següent:

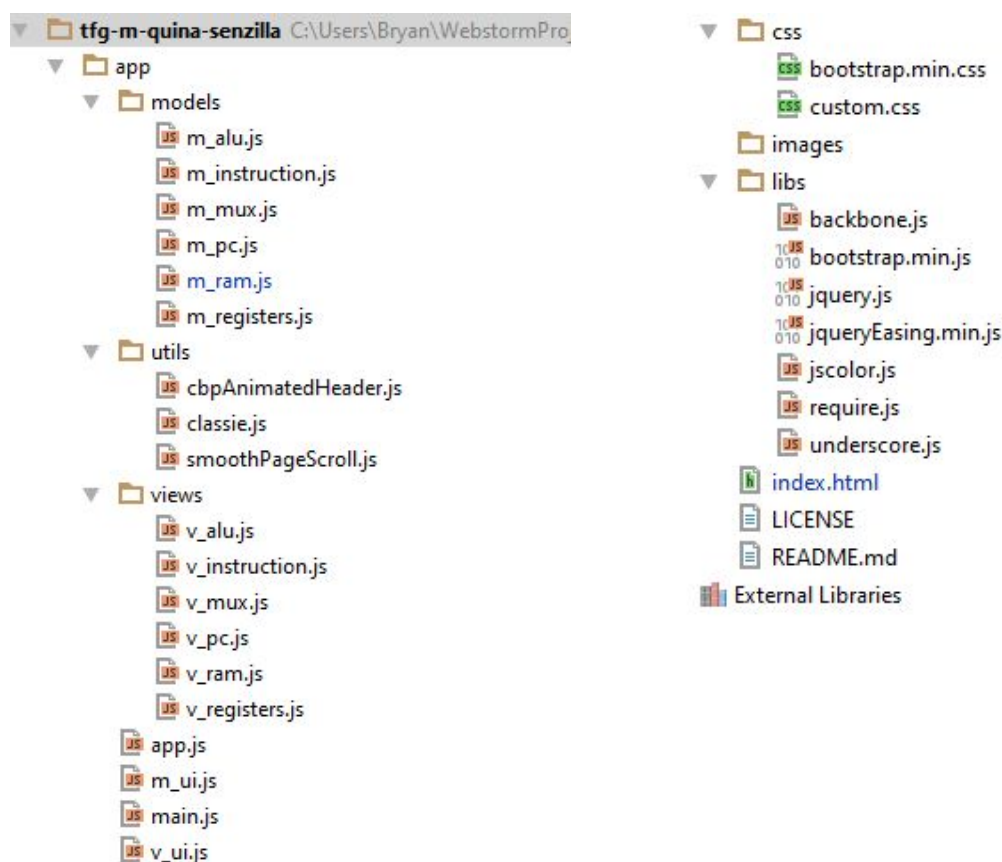


Figura 19: Estructura de directoris del projecte.

En l'arrel ens trobem el index.html que serà el primer fitxer que carregaran els usuaris del servidor web. Després tenim un fitxer d'informació sobre el projecte (README.txt) per oferir més informació del projecte als usuaris i finalment un fitxer de text amb la llicència sobre la qual està definit el projecte, en aquest cas una llicència GNU General public license versió tres.

Continuant amb els directoris que ens trobem en l'arrel veiem els següents:

- app: directori on trobarem el nucli de l'aplicació, aquesta carpeta separa els models i les vistes de Backbone, a més, de tenir una carpeta de funcions auxiliars (utils) que utilitzem pel disseny i configuració de la web.

En l'arrel d'aquest directori trobem els models i vistes més globals i les classes que comencen a executar el programa des de l'inici.

- css: aquí guardem tots els fitxers de disseny css que conté la nostra aplicació.

La nostra aplicació fa servir els fitxers de css de Bootstrap més un fitxer custom.css per sobreescriure i personalitzar encara més el disseny de la pàgina.

- images: és on es guarden totes les imatges de l'aplicació.
- libs: totes les llibreries i frameworks utilitzats per a l'aplicació.

9.2 Càrrega de llibreries

Per començar a desenvolupar un projecte en JavaScript necessitem estructurar l'aplicació i sobretot carregar correctament totes les llibreries i els frameworks que utilitzem perquè a l'hora d'executar la mecànica de l'aplicació, no hi hagi problemes de dependències.

Aquest va ser un dels primers problemes que em vaig trobar. El client descarregava asincronament les llibreries des del servidor i algunes dependències no s'executaven correctament.

Per solucionar-ho vaig utilitzar el framework RequireJs, que ens permet crear mòduls de les llibreries i carregar-les de forma ordenada.

```
require.config({
  shim : {
    underscore : {
      exports : '_'
    },
    bootstrap : {
      deps : [ 'jquery' ],
      exports: 'Bootstrap'
    },
    backbone : {
      deps : [ 'jquery', 'underscore' ],
      exports : 'Backbone'
    },
    colorpicker: {
      deps : [ 'jquery', 'bootstrap' ]
    }
  },
  paths: {
    jquery: '../libs/jquery',
    jqueryEasy: '../libs/jqueryEasing.min',
    bootstrap: '../libs/bootstrap.min',

    backbone: '../libs/backbone',
    underscore: '../libs/underscore',

    classie: 'utils/classie',
    cbpAnimatedHeader: 'utils/cbpAnimatedHeader',
    smoothPageScroll: 'utils/smoothPageScroll',

    jscolor: '../libs/jscolor'
  }
});
```

Figura 20: Fitxer de configuració de RequireJs.

A més ens permet definir el camí d'on carregarà la llibreria i les dependències que té, amb això podem assegurar que en els navegadors no ens donin errors per carregar abans una llibreria que un altre. Aquest és l'exemple del fitxer main.js que està representat en la imatge anterior i que és el fitxer de configuració de RequireJS.

Després inicialitzem l'aplicació carregant totes aquestes dependències. Aquesta ha estat la configuració del fitxer main.js amb la configuració del Require.js.



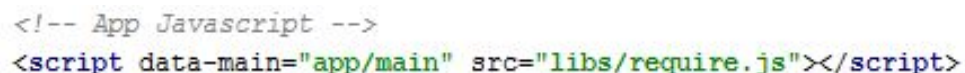
```
require(['jquery', 'jqueryEasy', 'bootstrap', 'classie', 'cbpAnimatedHeader', 'smoothPageScroll',
  'backbone', 'underscore', 'app', 'jscolor'], function($, jQuery, Bootstrap, Classie, CbpAnimatedHeader, SmoothPageScroll,
  Backbone, Underscore, App, Jscolor) {

  App.initialize();
  console.log("Todo cargado correctamente");

});
```

Figura 21: Fitxer de configuració de RequireJs on inicialitza l'aplicació un cop carregades les llibreries i frameworks.

Després amb definir al final del index.html l'entrada d'aquest fitxer ja ens aplicarà els canvis. Això ens aporta més claredat a més d'eliminar codi innecessari per declarar.



```
<!-- App Javascript -->
<script data-main="app/main" src="libs/require.js"></script>
```

Figura 22: Exemple implementació a index.html del framework RequireJS.

9.3 Estructura index.html “Responsive”

Un altre element important és el DOM (Document Object Model) de l’estructura del index.html, ja què mostrar els elements de la màquina senzilla no és una tasca fàcil, la informació s’ha de veure de forma clara i només tenim una pantalla i resolucions limitades per representar-la.

El resultat final està representat en el següent esquema:

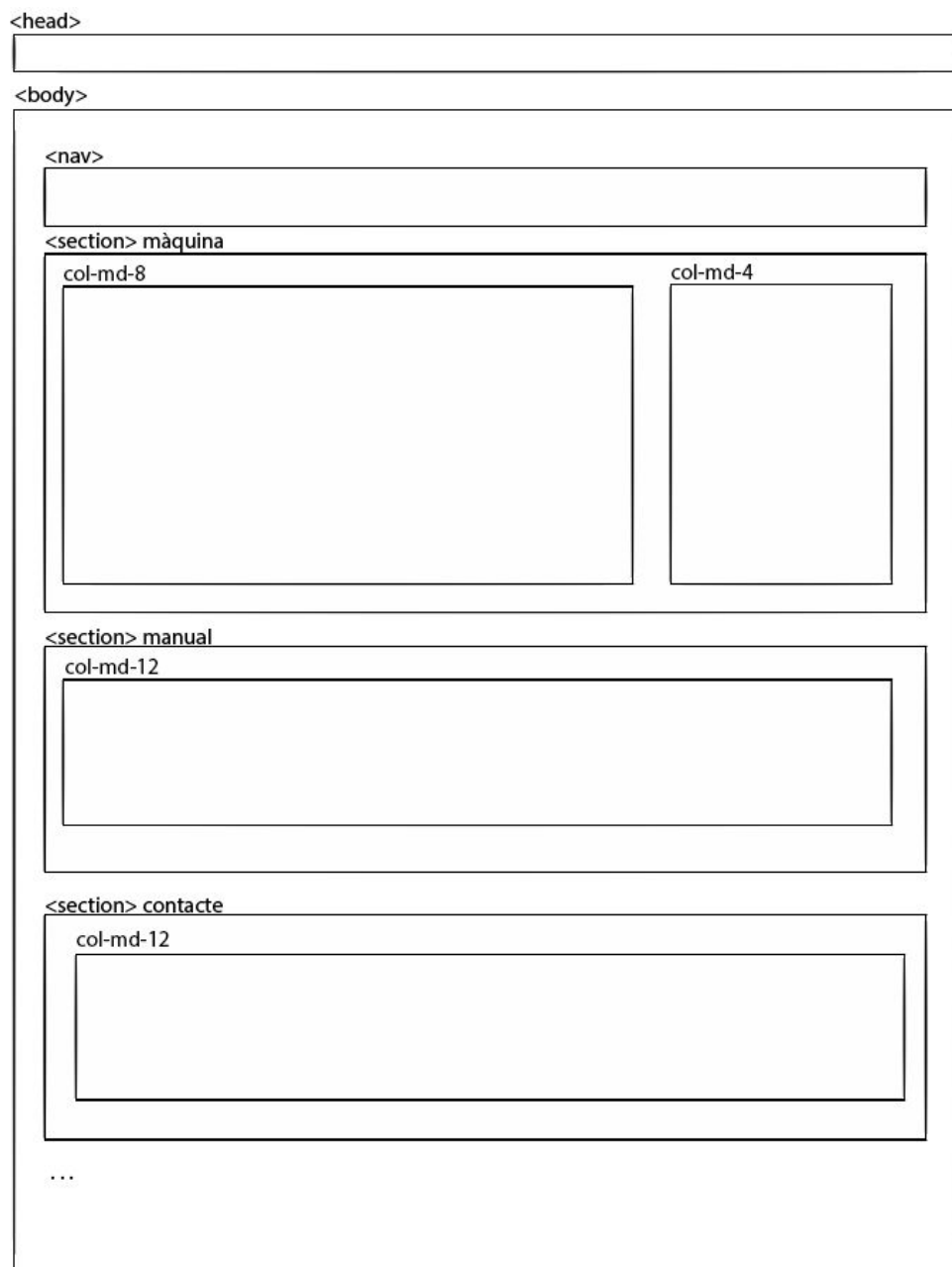


Figura 23: Esquema de l’estructura del DOM del projecte.

Aquesta estructura està composta a l'inici per un “head” on es defineixen les meta-dades de la pàgina, el títol, els links dels fitxers de disseny css i unes fonts concretes per millorar l'estètica de la pàgina.

```
<!DOCTYPE html>
<html lang="ca">

<head>

  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="Aplicació de la màquina senzilla per a la universitat de girona">
  <meta name="author" content="Bryan Pierre">

  <title>Simulador Màquina Senzilla</title>

  <!-- Bootstrap Core CSS -->
  <link href="css/bootstrap.min.css" rel="stylesheet">

  <!-- Custom CSS -->
  <link href="css/custom.css" rel="stylesheet">

  <!-- Custom Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css">
  <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700' rel='stylesheet' type='text/css'>

</head>
```

Figura 24: Fragment secció “head” del index.html.

Respecte al que és el “body”, ho hem estructurat per seccions, però abans de tot hi ha un menú per facilitar a l'usuari la navegació per la pàgina.

Aquest menú té uns quants efectes com el “smoth scroll” que ens porta per les diferents seccions de la pàgina amb un moviment suau o la il·luminació de la secció on ens trobem.

També és “responsive” que vol dir que, quan disminueix d'una determinada resolució, el menú es transforma a versió de mòbil per una visualització més còmode per l'usuari.

Figura 25: Representació del menú de l'aplicació en format d'escriptori.

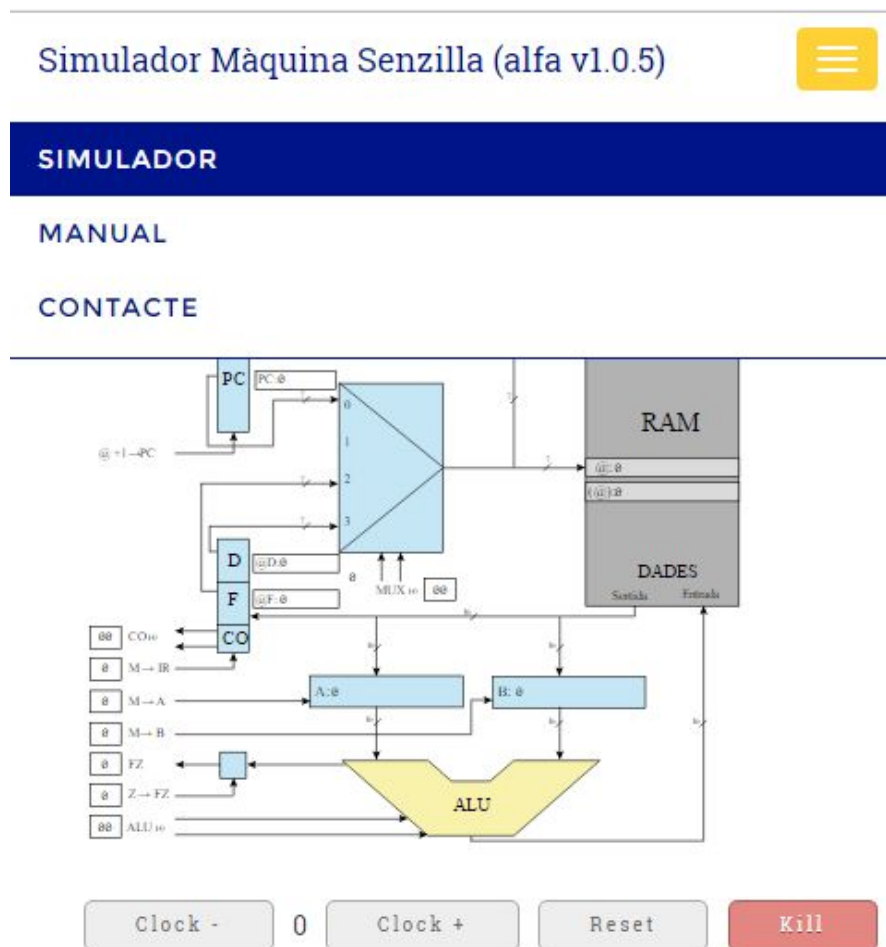


Figura 26: Representació de l'aplicació en format mòbil.

Continuant amb el body i les seccions que envolta, la primera secció és la més important, ja que allotja la imatge de la màquina senzilla, la RAM i la consola. Aquesta secció està separada en dues columnes que utilitzen les classes de disseny de Bootstrap.

Aquestes classes ens permeten dissenyar de forma que quan la resolució de la pantalla on està visualitzada l'aplicació disminueix s'adapten a aquesta de manera "responsive" això és molt important per donar suport a mòbils i tablets.

El funcionament concret és que bootstrap separa l'ample total en 12 columnes, i a partir d'allà podem fer les nostres separacions. Com surt a la imatge:

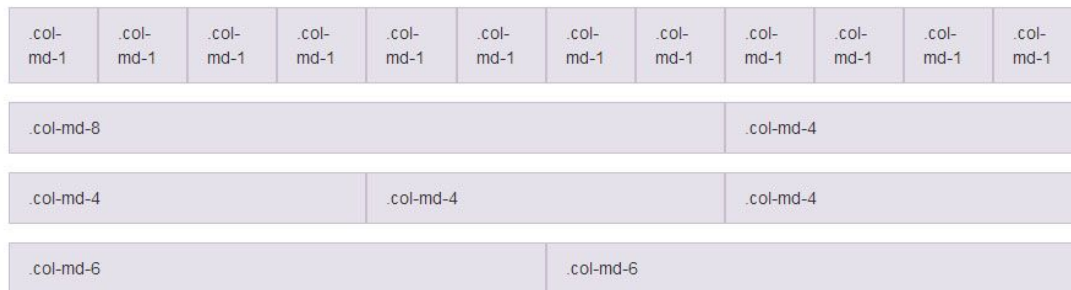


Figura 27: Model de separació de columnes de Bootstrap.

9.4 Models i vistes de l'aplicació

Un cop amb les llibreries carregades i una estructura al index.html falta desenvolupar tots els elements de la màquina senzilla.

Ho hem fet fent servir una metodologia orientada en el model de classes, separant cada element de la màquina senzilla en un objecte diferent, com hem mostrat anteriorment en l'apartat de diagrama de classes.

Per fer-ho vàrem crear un model i una vista per a cada element seguint el procediment que ens facilita el framework de Backbone.

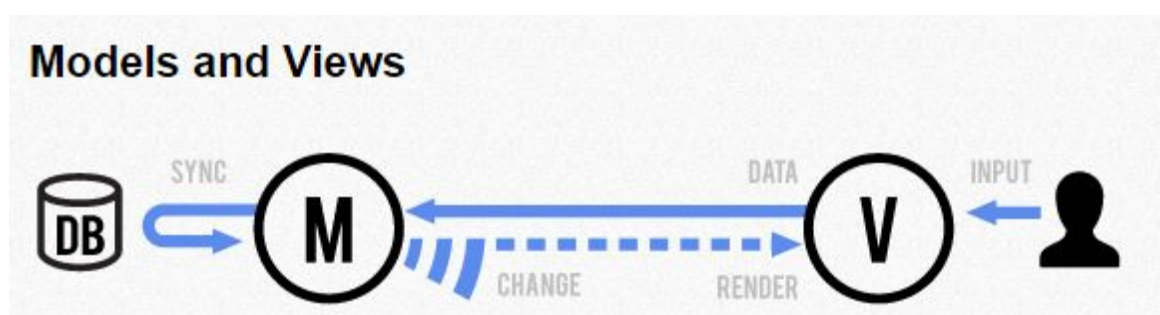


Figura 28: Esquema del model de classes de Backbone.

Com hem comentat anteriorment a l'apartat de descripció de Backbone, la vista i el controlador estan en el mateix fitxer, anomenats a l'aplicació com v_ "classe" i els models on es guarden les dades com m_ "classe". Aquest és un patró per diferenciar els tipus d'arxius.

Un exemple de model de l'ALU seria:

```

/**
 * Aquest model representa a l'ALU de la màquina senzilla. Guardarem els valors
 * dels operands destí i font per després fer les operacions corresponents.
 *
 * A més guardem els controls dels flags que activen a l'ALU i el valor del fz.
 */
define(['backbone', 'underscore'], function(Backbone) {

    var Alu = Backbone.Model.extend({

        defaults: {
            inputA: "0",          /**< Entrada de dades del registre A. */
            inputB: "0",          /**< Entrada de dades del registre B. */
            sortida: "0",         /**< Resultat de les operacions efectuades per l'ALU. */
            fz: "0",              /**< Valor del flag de zero en tot moment. */
            flag: ["0", "0"]      /**< Valor dels controladors que indiquen a l'ALU quina operació realitzar. */
        },

        /**
         * Aquesta funció fa la suma del contingut dels dos registres A i B.
         * Converteix les instruccions i les dades a binari i fa la suma.
         * Modifica el flag de Zero si el resultat és 0.
         *
         * @returns {number} Retorna -1 si s'ha passat dels bits màxims permesos per la RAM. Altrament 0.
         */
        add_inputs: function() {...},

        /**
         * Funció que mou directament el valor del registre B a la sortida de l'ALU.
         *
         * Modifica el flag de Zero si el valor del registre B és 0.
         */
        mov: function () {...},

        /**
         * Compara els valors dels registres A i B.
         *
         * Modifica el flag de Zero si els valors de A i B són iguals.
         */
        cmp: function () {...},

        /**
         * Converteix una instrucció o un enter a binari.
         *
         * @param input (String) Una instrucció en format (add 110, 113) o un enter en base decimal.
         * @returns {*} (Enter) Retorna un enter que és l'entrada transformada.
         */
        convertir_input: function (input) {...}

    });

    return Alu;
};

```

Figura 29: Fitxer Javascript que representa el model del element ALU de l'aplicació.

On es defineixen els seus atributs principals i les funcions que interactuen amb aquestes. Backbone també té una API per gestionar els models, per exemple té mètodes de “get” i “set” per obtenir i escriure sobre aquests atributs directament. Per tant ens estalvia moltes funcions de consulta i inserció.

L'exemple de vista de la mateixa classe ALU seria:

```

/**
 * Vista del model d'ALU, ens serveix per representar l'ALU
 * en la imatge de la màquina senzilla.
 */
define(['jquery', 'backbone', 'underscore', '../models/m_alu'], function($, Backbone, _, nalu) {

    var vAlu = Backbone.View.extend({

        /**
         * Inicialitza la vista de l'ALU
         */
        initialize: function(){
            _.bindAll(this, 'render');
            this.model = new nalu();
            this.render();
        },

        /**
         * Renderitza per defecte.
         */
        render: function() {
            var flag = document.getElementById("XMLID_338_");
            flag.innerHTML = this.model.get("flag").toString().replace(' ', '&nbsp;');

            var fz = document.getElementById("XMLID_334_");
            fz.innerHTML = this.model.get("fz");
        },

        /**
         * Renderitza els controladors de l'ALU.
         */
        render_control: function (senyal1, senyal0) {...}
    });
    return vAlu;
});

```

Figura 30: Fitxer Javascript que representa la vista del element ALU de l'aplicació.

On es representa com s'ha de mostrar tota la informació del model de l'ALU. En aquest cas no tenim controladors en aquesta classe, ja que els controladors estan assignats a una classe més general que llençarà les instruccions pertinents per reproduir les vistes de l'ALU i la resta d'elements.

9.5 Classes generals “app” i “ui”

Per controlar tots els models i vistes creats per a cada classe del projecte, es va crear una classe per emmagatzemar-les totes. És la classe Ui, que seria com la nostra interfície, que té el control de tots els esdeveniments que es fan a l'aplicació.

```
define(['backbone', 'underscore', 'views/v_ram', 'views/v_pc', 'views/v_instruction', 'views/v_mux', 'views/v_registers', 'vi

var Ui = Backbone.Model.extend({

  defaults: {
    ram: new v_ram(),
    pc: new v_pc(),
    instruction: new v_instruction(),
    mux: new v_mux(),
    registers: new v_registers(),
    alu: new v_alu(),

    state: 0,
    buffer: [],
    clock: 0,

    senyal1: '#90EE90',
    senyal0: '#FF6347',
    bus: '#00BFFF',

    assemblador: 0,

    diagrama: [ "-",0,0000000000000000,0,0,0000000000000000, "-",0,0000000000000000,0,0,0000000000000000,
      "-",0,0000000000000000,0,0,0000000000000000, "-",0,0000000000000000,0,0,0000000000000000,
      "-",0,0000000000000000,0,0,0000000000000000, "X,0,0000000000000000,0,0,0000000000000000"]
  },
});
```

Figura 31: Fitxer Javascript que representa el model la interfície UI.

El model de la classe Ui engloba les altres vistes de cada element de l'aplicació i serà des d'aquest model on s'executaran les funcions que facin funcionar la màquina senzilla.

```

/* =====

fetch: function() {...},

decodificacio: function () {...},

consultaFZ: function () {...},

operandB: function () {...},

operandA: function () {...},

execucio: function () {...},

/* =====

reset_clock: function () {...},

kill: function () {...},

/* =====

carregarInstruccioAnterior: function () {...},

get_instruccio: function () {...},

convertir_input:function (input) {...},

/* =====

render_normal: function () {...},

render_assembly: function () {...}

});

return Ui;

```

Figura 32: Fitxer Javascript que representa les funcions del model de la classe UI.

Aquestes funcions seran executades per la seva vista on es trobaran els controladors d'esdeveniments que estaran escoltant els canvis dels elements del DOM com els botons per executar les funcions.

```
define(['jquery', 'backbone', 'underscore', 'm_ui'], function ($, Backbone, _, v_ui) {

    var vUi = Backbone.View.extend({

        el: '#services',
        events: {
            "click #clock_plus" : "clock_plus",
            "click #clock_less" : "clock_less",
            "click #reset" : "reset_clock",
            "click #kill" : "kill",
            "click #save-colors" : "load_colors",
            "change input[name='fileInput']": "render_memoria_fitxer",
            "click #btnLaunch" : "editar_memoria",
            "click #btnSave" : "guardar_edicio_memoria",
            "click #assemblador" : "load_checkbox"
        }
    });
});
```

Figura 33: Esdeveniments que controla la vista de la classe Ui.

Finalment utilitzem una classe app per iniciar i separar el que és interfície i els elements del nucli de l'aplicació del fil inicial d'execució de l'aplicació.

```
define(['backbone', 'v_ui'], function (Backbone, ui) {

    var App = {};

    App.initialize = function () {
        var Fui = new ui();
    };

    return App;
});
```

Figura 34: Fitxer de configuració inicial de l'aplicació, on es comença l'execució de tot el nucli de l'aplicació.

9.6 Imatges SVG (Scalable Vector Graphics)

Aquest és un projecte molt visual i és la part més important de l'aplicació, ja que els usuaris que facin servir l'aplicació ho faran seguint sobretot els canvis que es produeixin en la imatge principal de la màquina senzilla.

Per això es va estudiar molt la manera de fer aquesta imatge, tant en la part d'accessibilitat com en la part de disseny.

La imatge final és la següent:

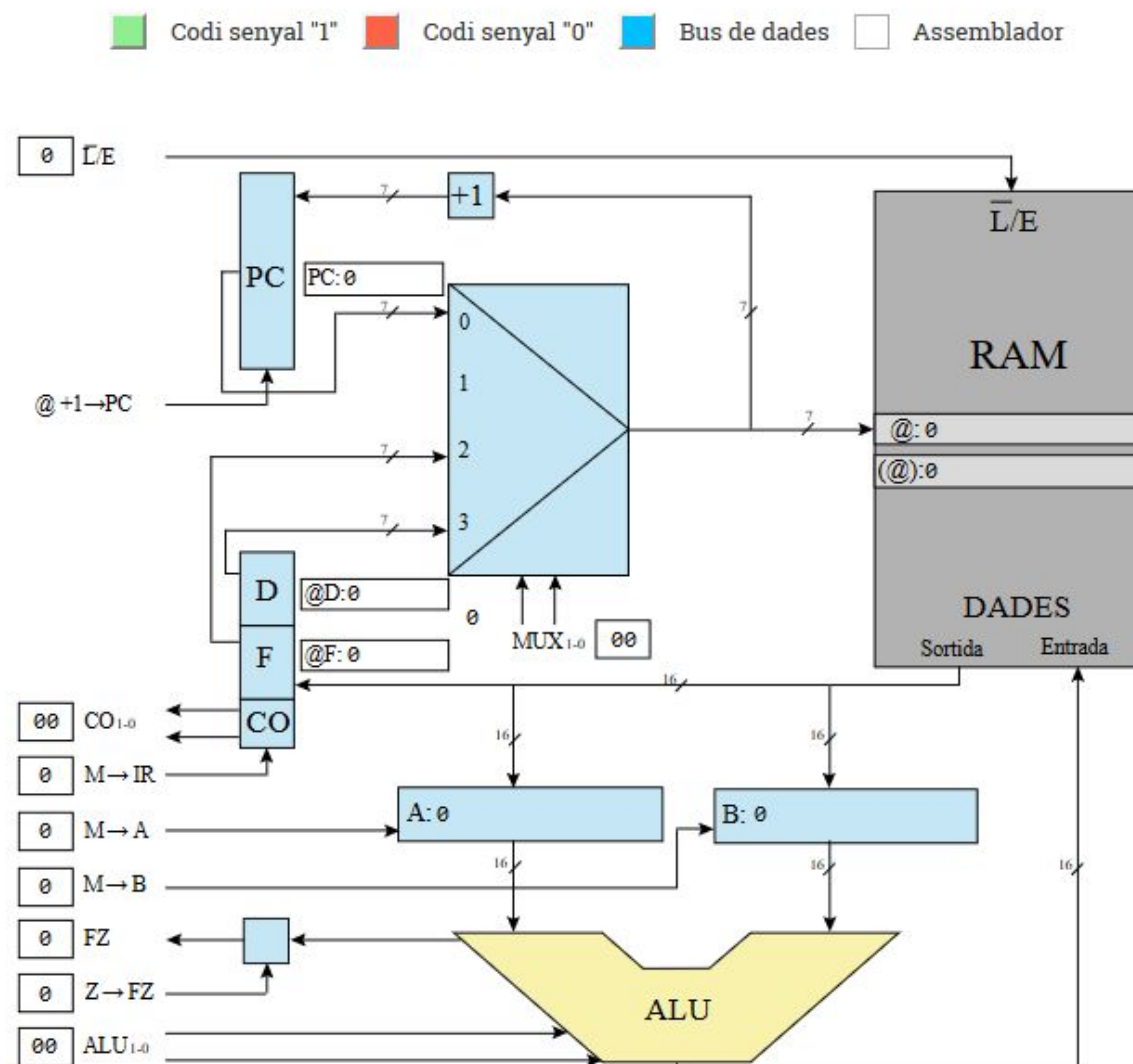


Figura 35: Imatge final de l'aplicació de la màquina senzilla.

Aquesta imatge es va fer a partir del programa Adobe Illustrator i es va exportar a SVG. La imatge està representada per vectors i classes generades pel programa, que per a cada vector ja li assigna un identificador dins de l'estructura.

Sobre les decisions més importants que s'han pres sobre la imatge:

- Representar directament a la imatge tots els valors del bus de control i de dades: va ser una decisió presa perquè l'antiga versió s'havia de fer un "on hover" per visualitzar segons quins valors. D'aquesta manera és visualment més fàcil d'interpretar i així també evitem aquesta funció no suportada en mòbils o tablets.

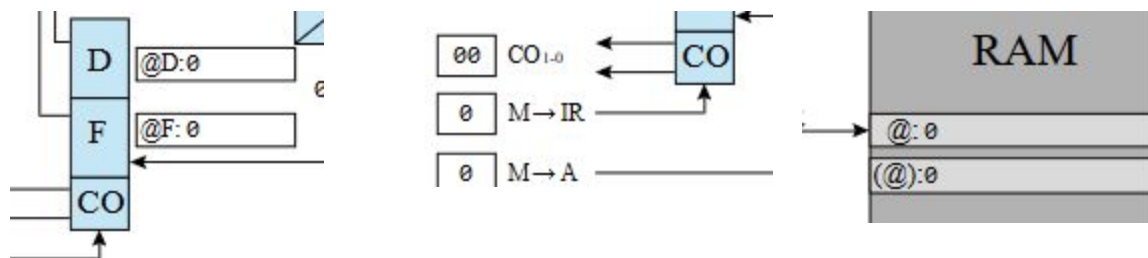


Figura 36: Exemple de la visualització del bus de control i dades de l'aplicació.

- Simplificació dels senyals de control: en l'antiga versió es colorejava de molts colors les representacions dels senyals de control de l'ALU, del multiplexador i del CO, ara s'ha separat en les unitats de control que li pertoqueu.

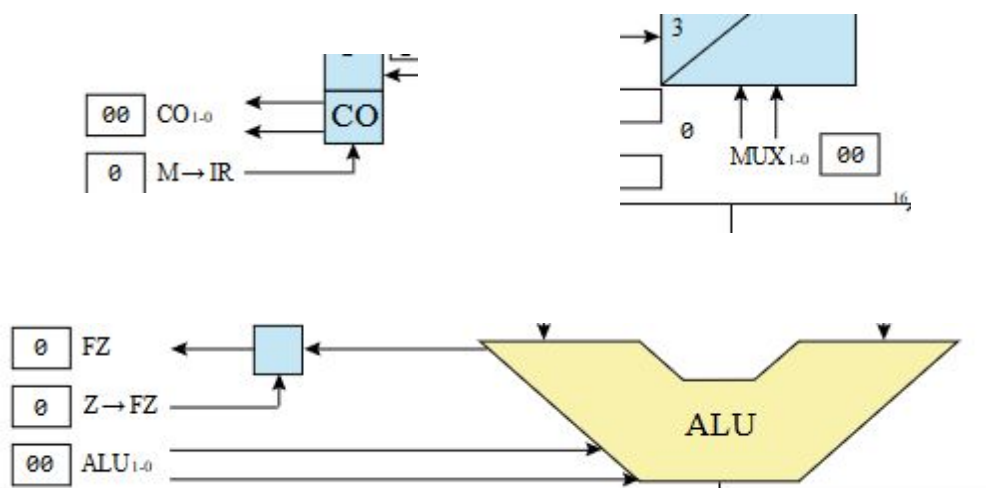


Figura 37: Entrada de doble bits per a l'ALU, multiplexador i CO del IR.

Pel que es tracta del tema d'accessibilitat, a part de mostrar els valors directament a la pantalla la imatge va canviant dinàmicament, per exemple:

- Animació en el flux de dades: per representar el flux de dades s'ha fet una animació de moviment en el sentit del flux.

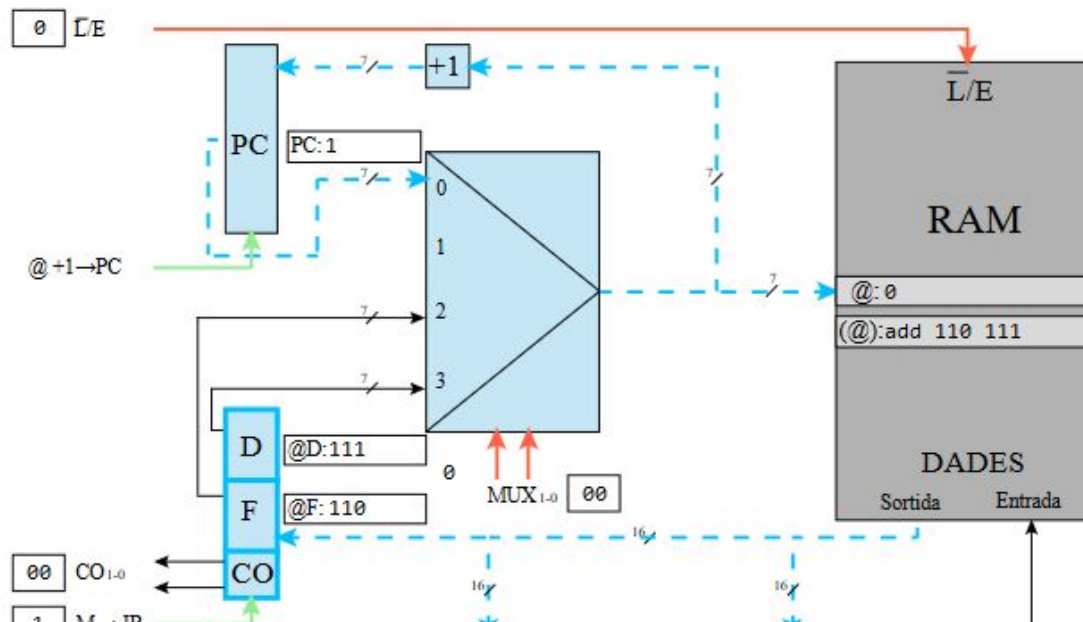


Figura 38: Visualització del flux de dades i els senyals de control en l'etapa de fetch.

- Augment en la grossor: també s'ha fet un augment en la grossor de les fletxes que s'utilitzen per visualitzar millor la diferència.
- Canvi de colors dels senyals de control i de flux: per garantir als usuaris un punt més d'accessibilitat s'ha fet servir una llibreria per canviar els colors dels senyals. Són els botons que es troben en la part superior de la imatge i que fan a l'hora de llegenda.

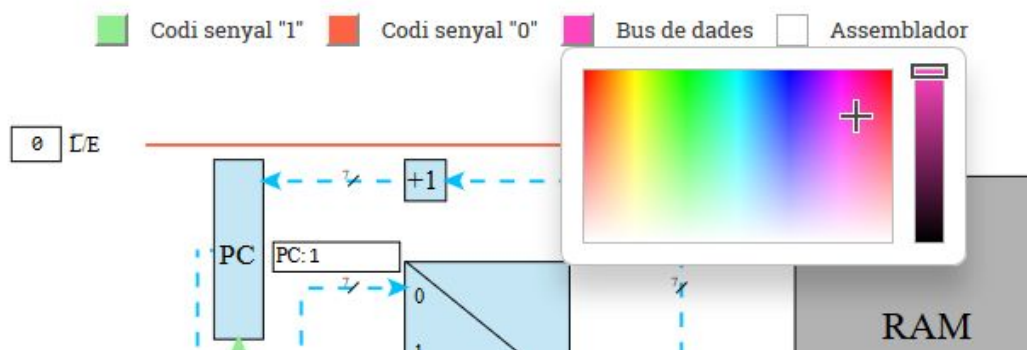


Figura 39: Paleta de colors per seleccionar el color del flux de dades i de control.

9.7 Representació de la memòria RAM

Per representar la memòria RAM, en un primer moment es va fer de la manera tradicional, una taula amb totes les adreces i un scroll per visualitzar-la sencera.

Després vàrem veure que no era gens còmode per llegir les instruccions que s'executaven i els valors que es guardaven a la memòria. Per això finalment es va optar per separar les instruccions de les dades.

Memòria

Etiqueta	Adreça	Instrucció
	0	add 110 111
	1	mov 111 113
	2	cmp 111 112
	3	beq 0 6
	4	cmp 110 110
	5	beq 0 0

Etiqueta	Adreça	Dades
	107	0
	108	0
	109	0
	110	1
	111	1
	112	3
	113	0

Examinar... exemple1.txt

Editar memòria

Figura 40: Visualització de la memòria RAM en dues parts: instruccions i dades.

També s'han afegit millores com la representació d'etiquetes, la selecció de la instrucció que estem executant i marcatge del valor que escrivim a la memòria.

A part aquesta nova versió de la memòria ens permet editar-la com si fos un llenguatge de programació sense limitacions.

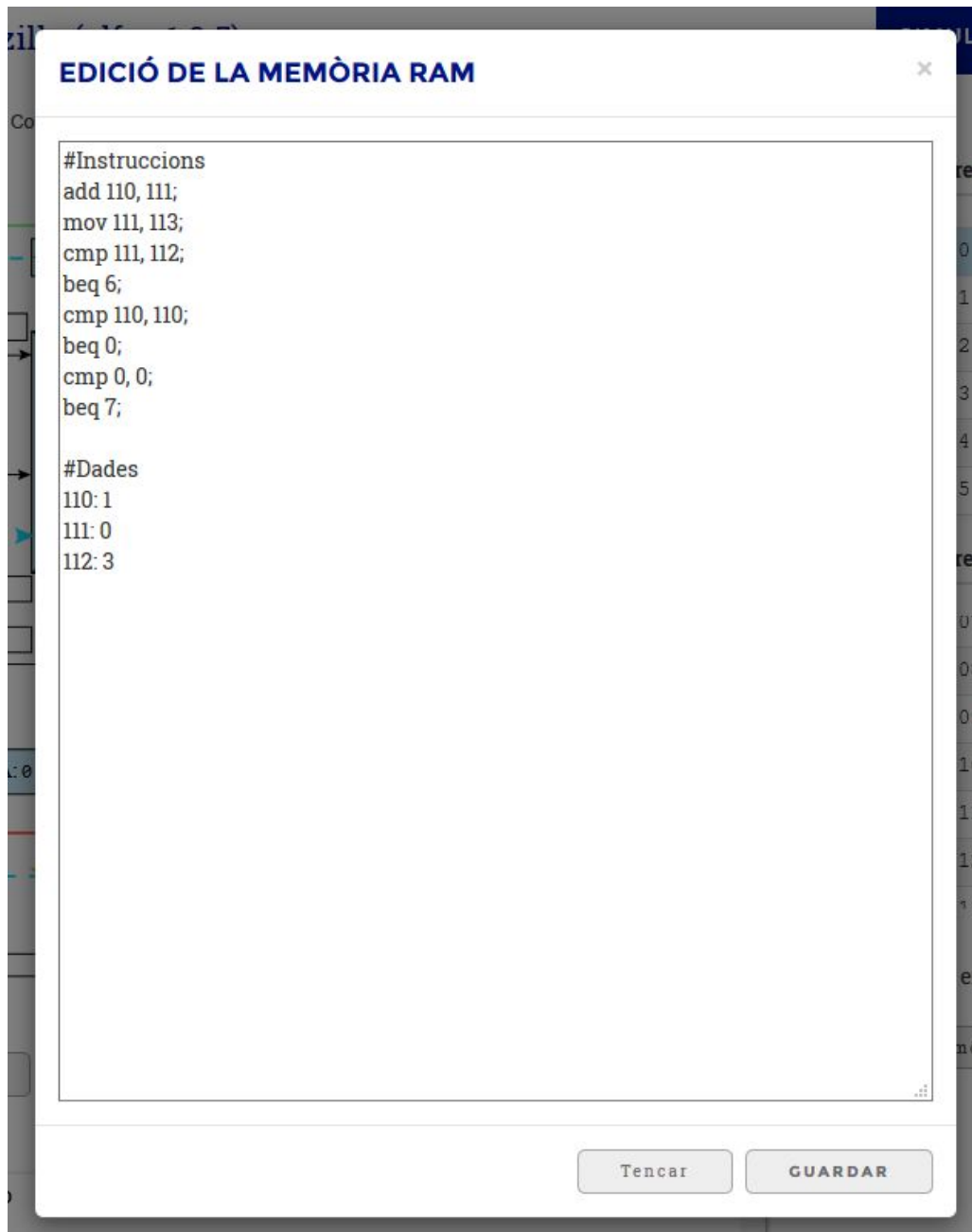


Figura 41: Panell d'edició manual de la RAM.

On gràcies al compilador o “parser” que hem creat podem programar com un llenguatge de programació, afegint comentaris, declarant les variables...

9.8 Consola

Per proposar a l'usuari de l'aplicació una eina més fiable d'interacció i seguiment de la màquina senzilla, es va dissenyar una consola per poder mostrar els esdeveniments de la màquina. Ja siguin per errors al carregar dades a la memòria, com missatges de confirmació, o el mateix flux de dades, tota interacció amb la màquina senzilla es representarà a la consola.

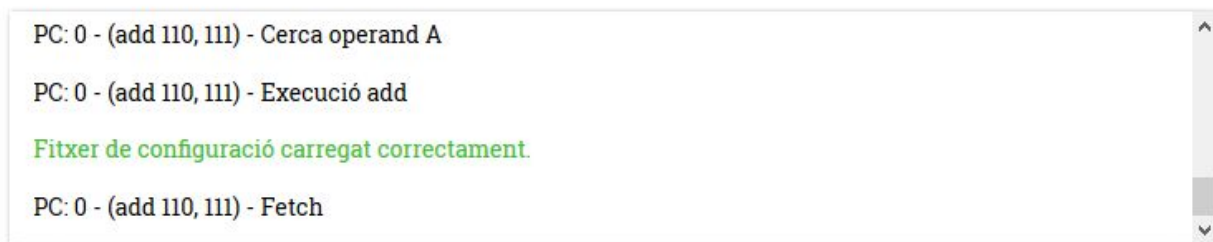


Figura 42: Consola de l'aplicació de la màquina senzilla.

9.9 Interpretació entrada de dades i algoritme “parsejador”

Una part important de l'aplicació és la seva capacitat de compilar els fitxers d'entrada i l'edició de la memòria.

En editar la memòria directament no hi ha problema, ja que quan es guarden les dades ja estan processades i directament es passa el text al parsejador.

El problema principal que varem tenir va ser l'entrada per fitxer, ja que la lectura de fitxer per Javascript utilitza funcions asíncrones. Per solucionar el problema es va fer ús d'un “Callback” per fer seqüencial aquesta funció de carregar el fitxer de text i després processar-la.

```
render_memoria_fitxer: function () {
    var self = this;
    var ram = this.model.get("ram");

    var llegirText = function (callback) { // Utilitzacio de Callbacks per llegir i processar sequencialment
        var fitxer = document.getElementById('fileInput');
        var file = fitxer.files[0];
        var textType = /\.text.*/;
        if (file.type.match(textType)) {
            var reader = new FileReader();
            reader.readAsText(file);
            reader.onload = function (event) {
                callback(event.target.result);
            }
        } else {
            self.afegir_log_consola("Format del arxiu no suportat! Ha de ser un .txt", 2);
        }
    };

    var processarText = function (text) {...};

    llegirText(processarText);
},
```

Figura 43: Exemple del de “Callback” utilitzat per llegir les dades de fitxer i processar-les seqüencialment.

Un cop tenim tot el contingut d'entrada, ja sigui per part de l'edició de la memòria com per l'entrada de fitxer, l'aplicació ho separarà línia per línia i les processarà amb una expressió regular que hem definit. Després segons la concordança amb aquesta funció es decideix guardar els valors, errors o ignorar.

L'expressió regular utilitzada és la següent:

regex=

```
^s*(?:((?:([a-z][a-z_]*)|(\d+))\s*:?)?s*(?:((?:([a-z][a-z_]*)|(\d+))\s*:(add|mov|cmp|beq)\s+(?:((?:([a-z][a-z_]*)|(\d+))\s*,\s*)?((?:([a-z][a-z_]*)|(\d+))|((?:\d+|x|".+"))\s*)+))?(#[^\n\r]*)?s*([\n\r;]*)/i
```

La nostra línia ens farà concordança si hi ha definida una etiqueta al davant, seguit de dos punts, després de les instruccions add,mov,cmp,beq, seguits de números o etiquetes de les adreces font i destí. A més de poder afegir comentaris amb el coixinet.

També ens farà concordança si només definim un valor numèric que serà interpretat com una dada.

L'algoritme que utilitzem per processar les línies és el següent:

Si és una instrucció

Si és BEQ

Si té destí

Si destí és una etiqueta

Si és una etiqueta definida al programa

Si la instrucció està etiquetada

GUARDA PRIMERA POSICIÓ LLIURE

Altrament

Si té adreça

Si està dins del rang de instruccions

GUARDA EN ADREÇA

Altrament

ERROR: Fora de rang

Altrament

GUARDA PRIMERA POSICIÓ LLIURE

Altrament

ERROR: Etiqueta no definida al programa

Altrament

Si la instrucció està etiquetada

GUARDA PRIMERA POSICIÓ LLIURE

Altrament

Si té adreça

Si està dins del rang de instruccions

GUARDA EN ADREÇA

Altrament

ERROR: Fora de rang

```

                                Altrament
                                GUARDA PRIMERA POSICIÓ LLIURE
Altrament
  Error: "Destí no definit"

Si és ADD,MOV,CMP
  Si té operador font
    Si té operador destí
      Si operador font és una etiqueta
        Si l'etiqueta operador font existeix
          Si operador destí és una etiqueta
            Si l'etiqueta operador destí existeix
              Si la instrucció té etiqueta
                GUARDAR
                Altrament
                  Si té adreça
                    Si està dins del rang
                      GUARDA
                      Altrament
                        ERROR
                      Altrament
                        GUARDA
                    Altrament
                      ERROR: Etiqueta destí no existeix
                  Altrament
                    Si la instrucció té etiqueta
                      GUARDAR
                      Altrament
                        Si té adreça
                          Si està dins del rang
                            GUARDA
                            Altrament
                              ERROR
                          Altrament
                            GUARDA
                        Altrament
                          ERROR: Etiqueta destí no existeix
                      Altrament
                        ERROR: Etiqueta font no existeix al programa
                  Altrament
                    Si operador destí és una etiqueta
                      Si l'etiqueta operador destí existeix
                        Si la instrucció té etiqueta
                          GUARDAR
                          Altrament
                            Si té adreça
                              Si està dins del rang
                                GUARDA
                                Altrament
                                  ERROR
                              Altrament
                                GUARDA
                            Altrament
                              ERROR: Etiqueta destí no existeix
                        Altrament
                          Si la instrucció té etiqueta
                            GUARDAR
                            Altrament
                              Si té adreça

```



```

                                Si està dins del rang
                                GUARDA
                                Altrament
                                ERROR
                                Altrament
                                GUARDA
                                Altrament
                                ERROR: Etiqueta destí no existeix
                                Altrament
                                ERROR: No hi ha operador destí.
                                Altrament
                                ERROR: No hi ha operador font.
                                Altrament
                                Error: "Aquesta instrucció no pertany al repertori"

Si és una dada
    Si hi ha algun valor
        Si té etiqueta
            GUARDA
        Si té adreça
            Si la posició està ocupada
                ERROR: Posició ocupada
            Altrament
                Si està dins del rang de dades
                    GUARDA
                Altrament
                    ERROR: Fora de rang
        Altrament
            GUARDA
    Altrament
        IGNORAR

Altrament
    Error: "Format incorrecte"
```

10. Implantació i resultats

10.1 Implantació del Servidor Web

Un cop tenim tota la part del client definida ens cal fer la part del servidor, hem d'hostejar l'aplicació en un servidor obert a Internet perquè els usuaris puguin utilitzar-la.

Aquesta és una segona part del projecte que correspon més a l'àrea d'administradors de sistemes però que és igual d'important que la de programar el client.

Per això vàrem decidir utilitzar els serveis d'Amazon Web Service per implantar el servidor.



Figura 44: Logotip d'Amazon Web Services

Amazon Web Services ens proporciona un any de serveis gratuïts. El que vol dir que posa a la nostra disposició instàncies de sistemes operatius en màquines seves. Això sí molt limitades respecte al seu hardware.

La instal·lació és bastant senzilla, ja que un cop et fas un compte a la seva web podem accedir a un panell interactiu on ens donen la possibilitat d'entrar a EC2 (Elastic compute Cloud) per crear i configurar un servidor.

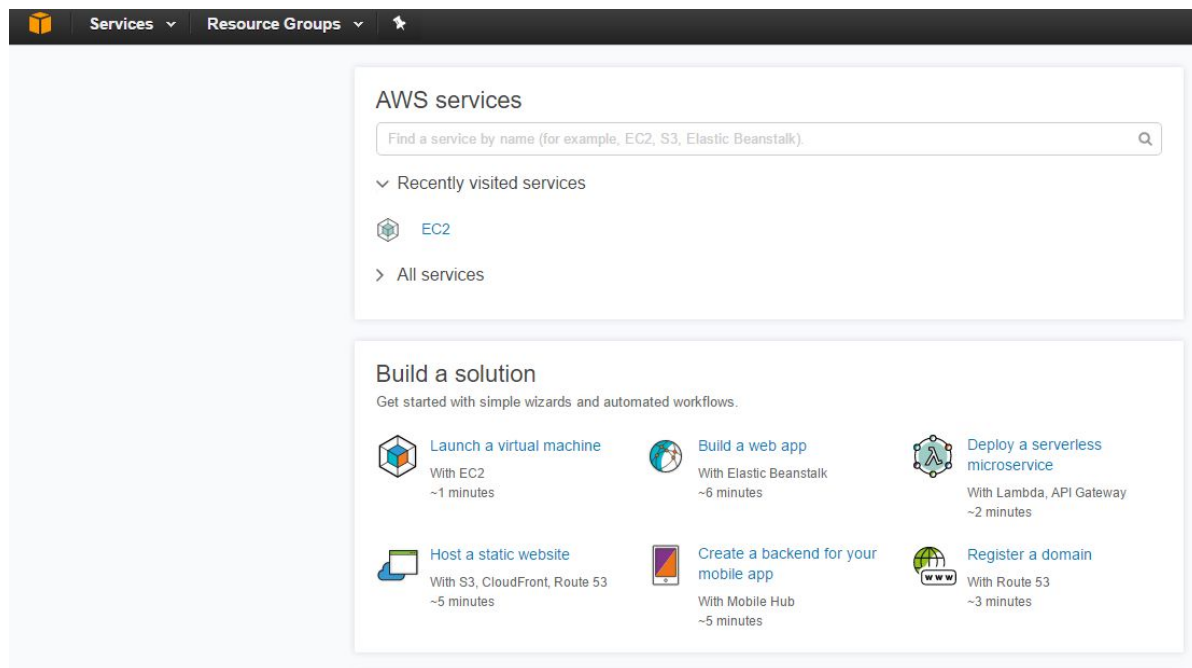


Figura 45: Panell principal d'AWS Services.

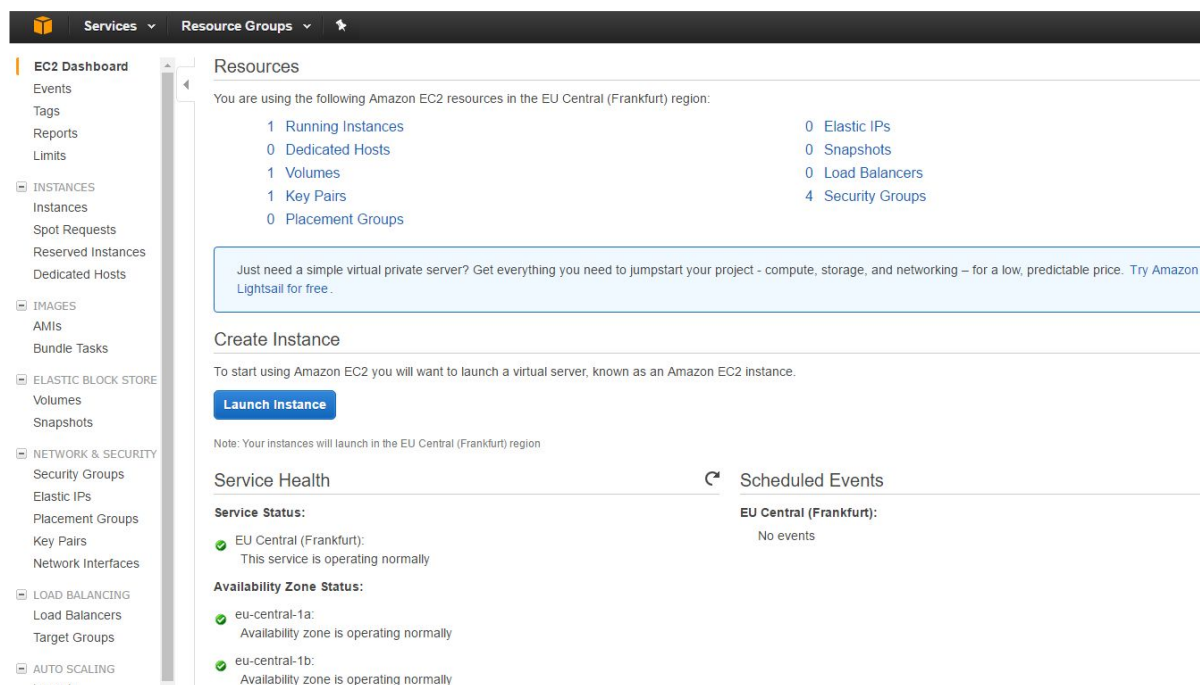


Figura 46: Panell principal del creador i configurador d'instàncies EC2 de Amazon Web Services.

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
	i-99033525	t2.micro	eu-central-1b	running	2/2 checks...	None	ec2-35-156-50-86.eu-ce...	35.156.50.86	ubuntu-server	disabled

Instance: **i-99033525** Public DNS: **ec2-35-156-50-86.eu-central-1.compute.amazonaws.com**

Description Status Checks Monitoring Tags

Instance ID	i-99033525	Public DNS	ec2-35-156-50-86.eu-central-1.compute.amazonaws.com
Instance state	running	Public IP	35.156.50.86
Instance type	t2.micro	Elastic IPs	
Private DNS	ip-172-31-21-13.eu-central-1.compute.internal	Availability zone	eu-central-1b
Private IPs	172.31.21.13	Security groups	web security - view inbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-6f642a06	AMI ID	ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20161020 (ami-8504fdea)
Subnet ID	subnet-10185f6b	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	ubuntu-server
EBS-optimized	False	Owner	105992125092
Root device type	ebs	Launch time	November 9, 2016 at 3:30:04 PM UTC+1 (1181 hours)
Root device	/dev/sda1	Termination protection	False
Block devices	/dev/sda1	Lifecycle	normal
		Monitoring	basic
		Alarm status	None
		Kernel ID	-

Figura 47: Panell administració d'instàncies amb detall.

Aquí veiem que ja està la instància creada, a més de mostrar més dades com les DNS, la IP pública l'estat de la màquina....

Per crear aquesta instància es va seleccionar una AMI d'Amazon, que són els sistemes operatius, amb aplicacions i serveis que es necessiten per crear la instància. Amazon proporciona moltes AMI's amb diferents sistemes operatius. Jo en aquest cas vaig escollir un Ubuntu Server.

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your c

Quick Start

My AMIs

AWS Marketplace

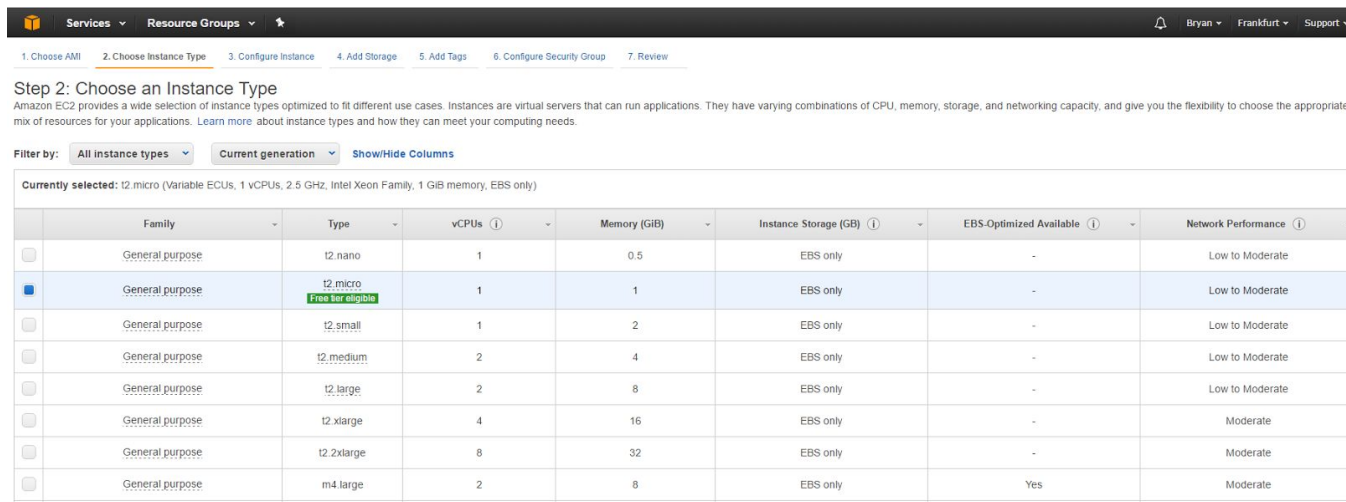
Community AMIs

Free tier only 1

Quick Start	AMI	AMI ID	Root device type	Virtualization type
Amazon Linux	Amazon Linux AMI 2016.09.1 (HVM), SSD Volume Type	ami-211ada4e	ebs	hvm
Red Hat	Red Hat Enterprise Linux 7.3 (HVM), SSD Volume Type	ami-e4c63e8b	ebs	hvm
SUSE Linux	SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type	ami-c425e4ab	ebs	hvm
Ubuntu	Ubuntu Server 16.04 LTS (HVM), SSD Volume Type	ami-b03fedf	ebs	hvm
Windows	Microsoft Windows Server 2016 Base	ami-425d9c2d	ebs	hvm

Figura 48: Panell de selecció d'AMI's d'Amazon.

Un cop es selecciona el tipus de sistema operatiu s'ha d'escollir el hardware que es vol. En aquest cas com hem comentat abans, el hardware que ofereixen gratuït són les instàncies del tipus “micro” que tenen pocs recursos.



Step 2: Choose an Instance Type
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

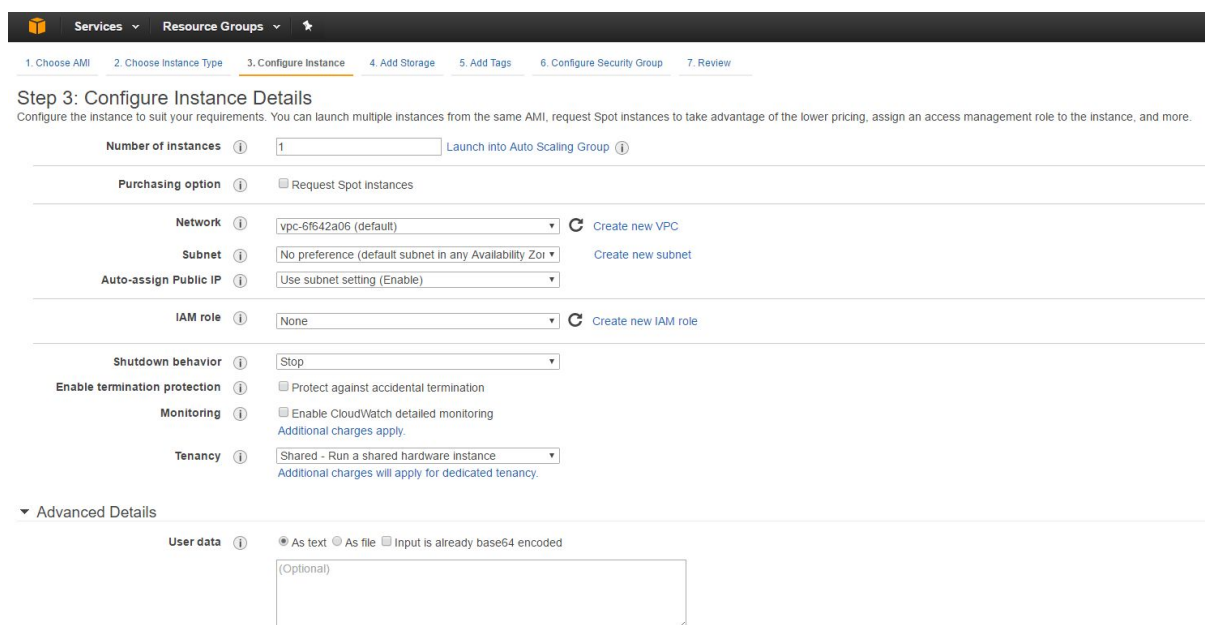
Filter by: **All instance types** **Current generation** [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate

Figura 49: Llista dels tipus d'instàncies segons recursos hardware d'Amazon.

Després podem configurar el tipus de xarxa que volem i les subxarxes, fins i tot el rang. En aquest punt es va deixar per defecte.



Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances 1 [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network vpc-6f642a06 (default) [Create new VPC](#)

Subnet No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP Use subnet setting (Enable)

IAM role None [Create new IAM role](#)

Shutdown behavior Stop

Enable termination protection ☐ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring
[Additional charges apply.](#)

Tenancy Shared - Run a shared hardware instance
[Additional charges will apply for dedicated tenancy.](#)

Advanced Details

User data ☒ As text ☐ As file ☐ Input is already base64 encoded
(Optional)

Figura 50: Configuració de la xarxa de la instància.

El següent pas és la capacitat d'emmagatzematge que volem, que a més d'especificar els directoris de root ens permet crear més particions dins del sistema.

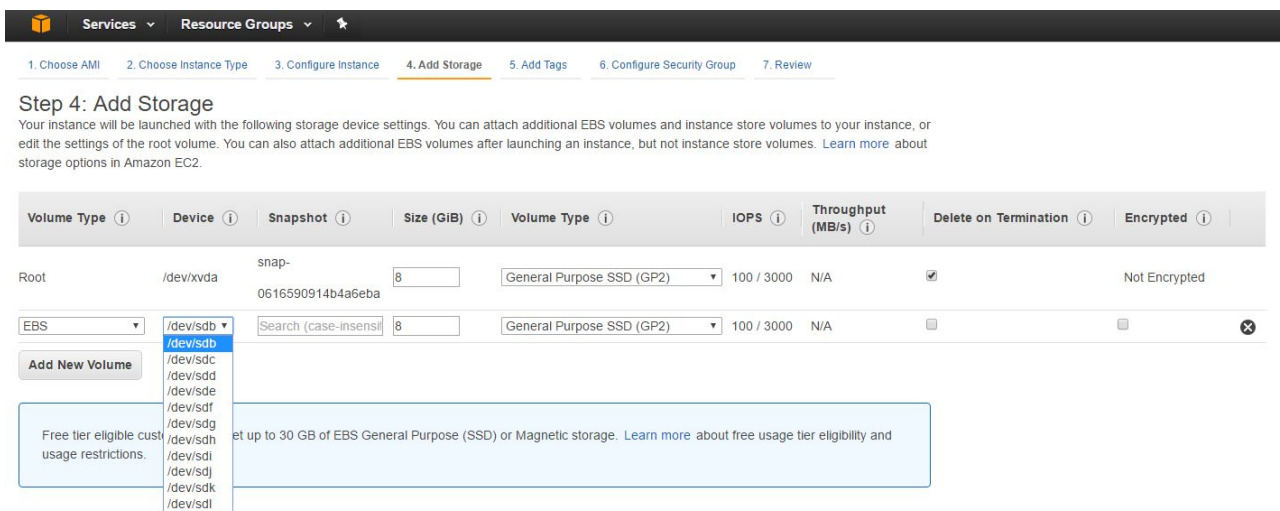


Figura 51: Configuració i selecció de particions.

Per a aquesta configuració es va deixar per defecte. El següent pas és el d'afegir "Tags" a la instància però no es va fer perquè no és necessari.

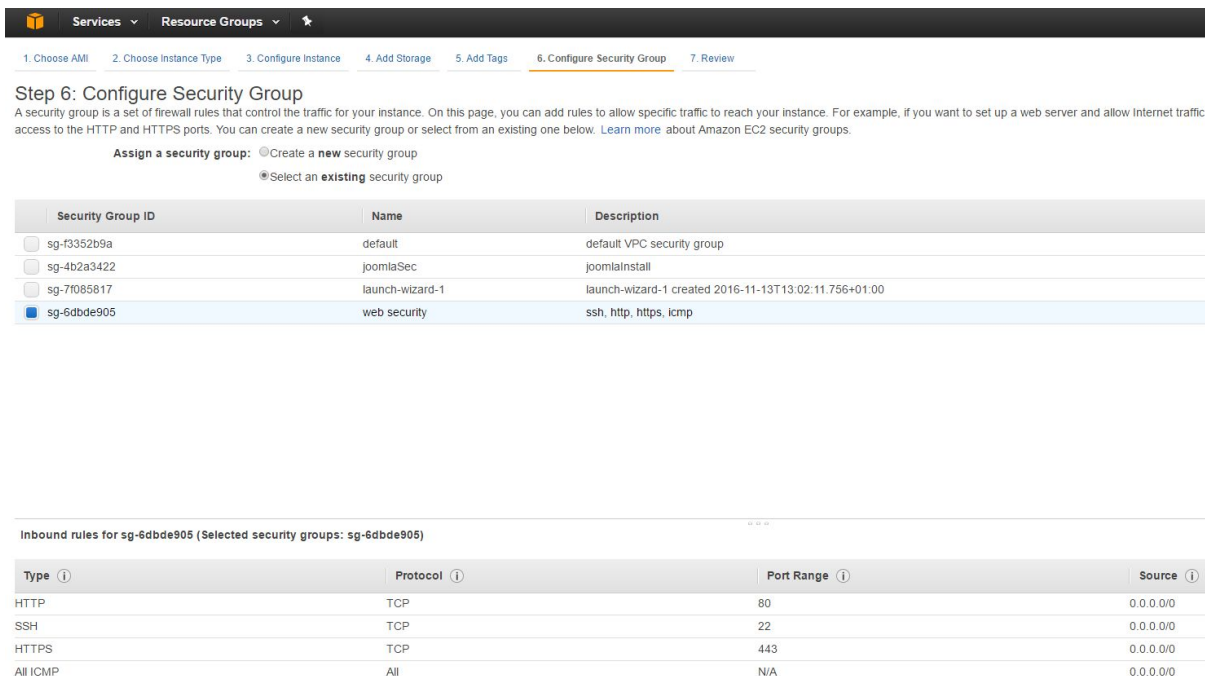


Figura 52: Configuració del firewall del servidor.

Aquest últim apartat de configuració és important perquè definim el “firewall” del nostre servidor. En aquest cas s’ha deixat només accés al port 80 http, 443 https, 22 ssh i que es pugui fer pings (icmp).

Un cop configurat apliquem els canvis i Amazon ens crearà la instància i ens generarà una clau “.pem” per connectar-nos via ssh amb el servidor.

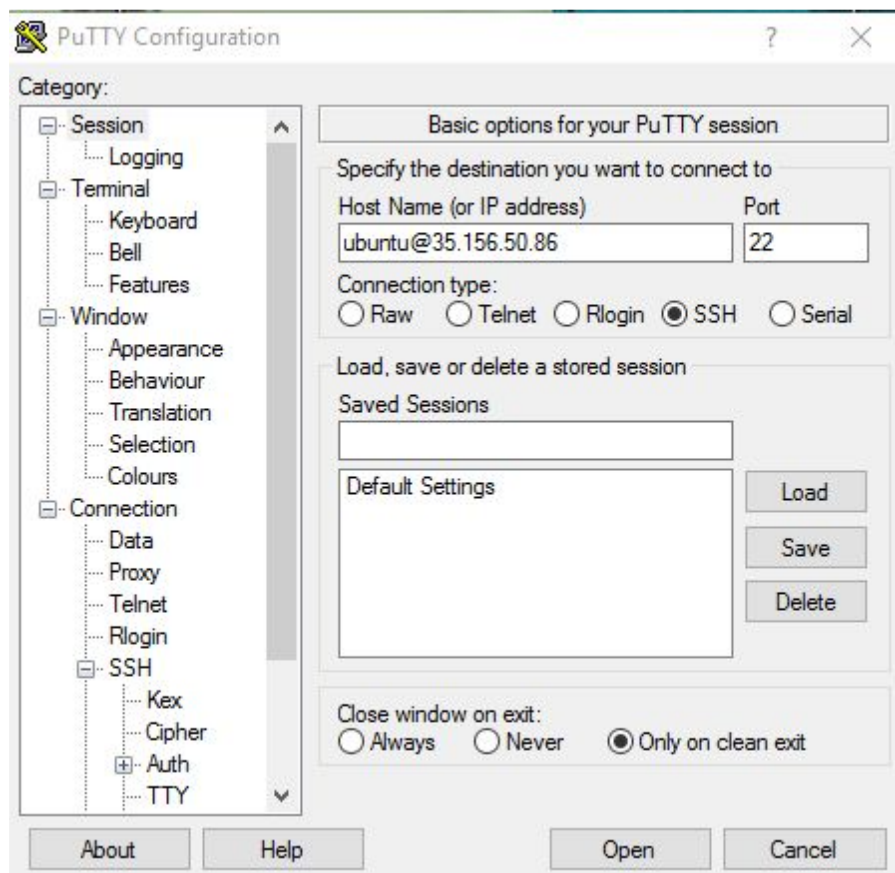
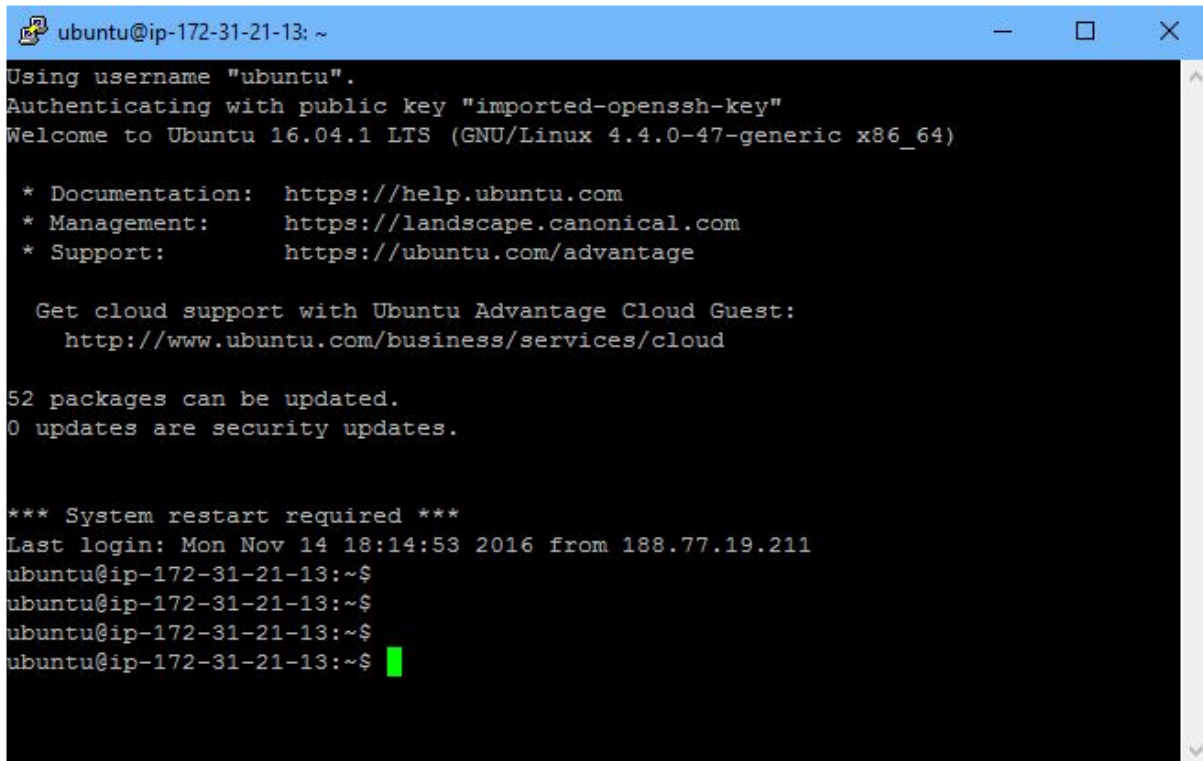


Figura 53: Aplicació Putty per realitzar connexions ssh des de Windows.

Ja només ens cal connectar-nos al servidor amb algun programa com Putty per a Windows o directament per ssh en si ho fem en un sistema Linux.

A terminal window titled 'ubuntu@ip-172-31-21-13: ~' with standard window controls. The terminal output shows an SSH session: 'Using username "ubuntu".', 'Authenticating with public key "imported-openssh-key"', 'Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-47-generic x86_64)', followed by links for documentation, management, and support. It then reports '52 packages can be updated.' and '0 updates are security updates.' A message states '*** System restart required ***' and shows the 'Last login' time. The prompt 'ubuntu@ip-172-31-21-13:~\$' is shown four times, with a green cursor at the end of the last one.

```
ubuntu@ip-172-31-21-13: ~
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

52 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Mon Nov 14 18:14:53 2016 from 188.77.19.211
ubuntu@ip-172-31-21-13:~$
ubuntu@ip-172-31-21-13:~$
ubuntu@ip-172-31-21-13:~$
ubuntu@ip-172-31-21-13:~$
```

Figura 54: Connexió establerta per ssh al nostre servidor d'Amazon.

Un cop ja amb el servidor en marxa hem de configurar un servidor Web, en aquest cas Apache2.

Per instal·lar-lo farem servir la comanda:

sudo apt-get install apache2

Un cop instal·lat, no ens queda res més que pujar el projecte en el directori arrel on l'Apache escolta, ja que la configuració inicial del Apache ens permet carregar una pàgina en aquest espai sense fer cap altre tipus de configuració.

Per això he fet ús d'un programa de transferència de fitxers ftp, el Filezilla. He pujat els fitxers al directori `var/www/html`.

Ja finalitzada la transferència, l'aplicació de la màquina senzilla està oberta per a tothom en l'adreça del servidor d'Amazon.

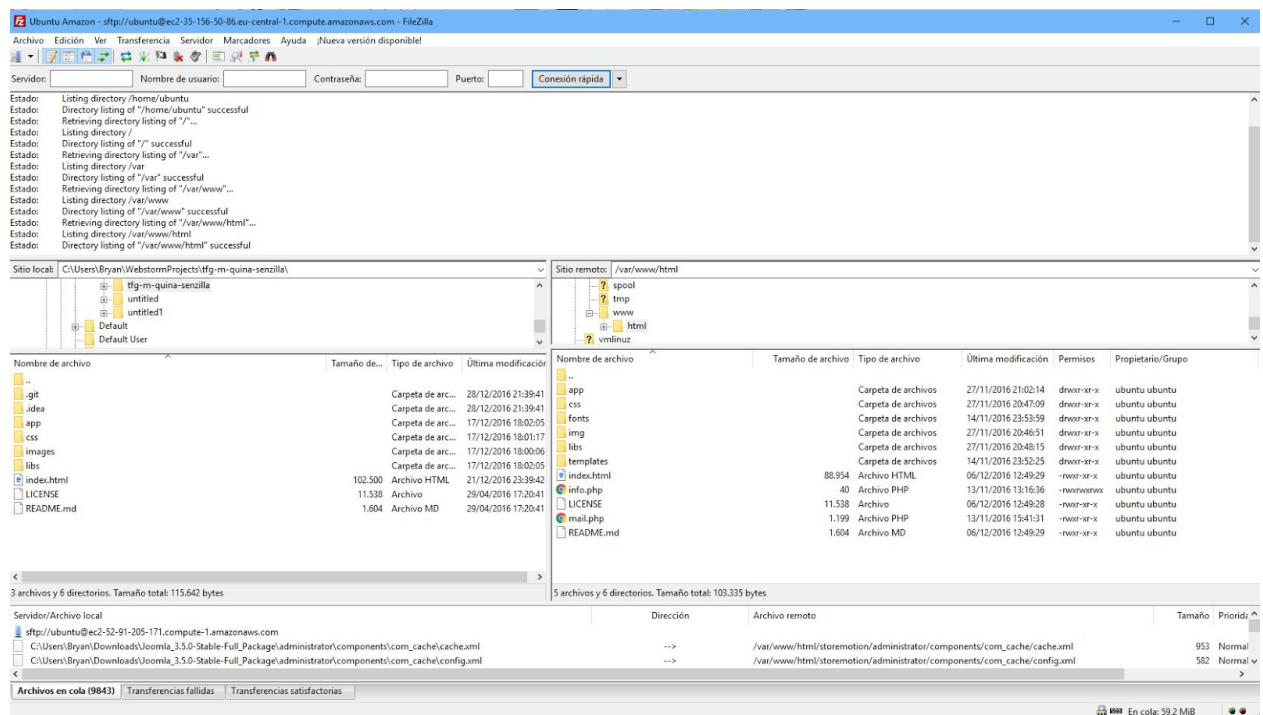


Figura 55: Trasnferència del projecte al servidor via Filezilla.

Per obtenir tot el codi font s'ha fet un repositori públic a GitHub:

<https://github.com/simuladormaquinasenzilla/SimuladorMaquinaSenzilla>

10.2 Fase beta de l'aplicació

Aquest muntatge en els servidors d'Amazon s'ha fet principalment per permetre a l'usuari utilitzar l'aplicació i detectar errors en la fase de desenvolupament del projecte.

Aquest període de proves ha solucionat molta part dels errors de l'aplicació i ha aportat altres punts de vista, a part del meu i dels tutors, que han beneficiat al desenvolupament de l'aplicació.

Per mantenir un control sobre aquest report, s'ha afegit al projecte un correu electrònic de contacte temporal i una llista de versions i modificacions que s'han fet en cada versió del programa.

Simulador Màquina Senzilla (alfa v1.0.5)

SIMULADORMANUALCONTACTE

CONTACTE

Bryan Pierre
ul9065209@campus.udg.edu

LLISTAT DE BUGS REPORTATS:

V1.0.1 BUG FIX

- [Entrades dades] No deixava afegir una etiqueta sense instrucció al darrere. Ara es pot fer i serà considerat una instrucció amb valor "add 0 0".
- [Etiquetes] Ara l'aplicació resol correctament les instruccions amb etiquetes. Les converteix en l'adreça corresponent al carregar-la al IR.

V1.0.2 BUG FIX

- [Visualització] El senyal de Z->FZ no es pintava correctament quan li tocava.
- [Visualització] S'ha tret el remarcatge del requadre de FZ quan s'executava la fase de descodificació.

V1.0.3 UPDATE

- [Memòria Ram] S'ha separat la memòria RAM en dues parts per una còmode visualització i utilització, a part es marca les instruccions que s'estan executant i quan es guarda un resultat a memòria.
- [Personalització] S'ha afegit una llegenda amb els colors de representació dels senyals i el bus, també és editable per si es vol canviar el color.

V1.0.4 BUG FIX

- [Personalització] S'ha canviat les llibreries del seleccionador de colors, donaven problemes en carregar i no suportava el tàctil de tablets i mòbils.

V1.0.5 UPDATE

- [Visualització] S'ha afegit un mode de vista "assemblador".

Figura 56: Contacte i llista de reports de cada versió de l'aplicació.

11. Conclusions

Els objectius del projecte eren la renovació de la Màquina Senzilla programada des de zero, a més d'adaptar-la a les noves tecnologies i fer-la més còmoda pels usuaris.

Els resultats obtinguts han sigut els següents:

- S'ha creat l'aplicació en JavaScript que funciona correctament i es pot utilitzar per a la docència de l'assignatura.
- L'entorn web és simple i intuïtiu, a més de ser “responsive” i adaptable tant per a plataformes tablets com a mòbils.
- El tema d'accessibilitat s'ha remarcat en alguns aspectes sobretot en les imatges per fer que l'aplicació sigui més fàcil de seguir.

Per a tot això que s'ha fet considero que s'han assolit els objectius marcats a l'inici del projecte.

Per altra banda, la meva opinió personal sobre el projecte en general és satisfactòria.

Durant totes les fases del projecte he anat evolucionant, ja que prèviament no havia fet cap pàgina web des de zero, sense per exemple un gestor de continguts.

Ha sigut això, la recerca de frameworks, eines i la seva aplicació directa que m'ha fet créixer i aprendre a desenvolupar projectes d'aquests tipus, on finalment, he agafat experiència en aquest camp.

Sobretot remarcar l'experiència obtinguda en programar Javascript i la utilització del framework Backbone que m'han ajudat a estructurar i dur a terme el projecte fins al final.

12. Treball Futur

Com a cada projecte sempre es pot estendre i afegir noves funcionalitats. Per a modificacions futures s'ha pensat en:

- Personalitzar més la visualització de la màquina senzilla, permetent als usuaris poder gestionar per ells mateixos els colors, la mida, la font... de la màquina senzilla.
- Afegir algun diagrama més de la màquina senzilla.
- Fer l'aplicació multi idioma, pels usuaris no parlants del català puguin gaudir de l'aplicació en el seu idioma.
- Fer millores i aportar més informació en els algoritmes programats.
- Implementar un sistema de mòduls que pugui ampliar i modificar l'estructura pròpia de la màquina senzilla, per exemple afegir més registres, més memòria, etc.

13. Bibliografia

- [1] Angular(2016). Recollit de <https://angular.io/>
- [2] Backbone(2010). Recollit de <http://backbonejs.org/>
- [3] Ember(2016). Recollit de <http://emberjs.com/>
- [4] JQuery(2016). Recollit de <https://jquery.com/>
- [5] jsColor(2017). Recollit de <http://jscolor.com/>
- [6]Apache2(sense data). Recollit de <https://wiki.apache.org/httpd/>
- [7] SPA (sense data). https://es.wikipedia.org/wiki/Single-page_application
- [8] RequireJS(sense data). Recollit de <http://requirejs.org/>
- [9]Bootstrap(sense data). Recollit de <http://getbootstrap.com/>
- [10] Underscore.js(2010). Recollit de <http://underscorejs.org/>
- [11] TeamGantt(sense data). Recollit de <https://www.teamgantt.com/>
- [12] Callbacks (2011). <https://fernetjs.com/2011/12/creando-y-utilizando-callbacks/>
- [13] SVG(sense data). http://www.w3schools.com/graphics/svg_intro.asp
- [14] Wikipedia. Recollit de <https://es.wikipedia.org>
- [15] GNU(2007). GNU Public license v3. <https://www.gnu.org/licenses/gpl-3.0.en.html>

14. Annexos

14.1 Índex de totes les imatges que apareixen en el projecte

<i>Figura 1: Esquema de la metodologia emprada al projecte.</i>	9
<i>Figura 2: Diagrama de Grantt del temps invertit en el projecte.</i>	11
<i>Figura 3: Concepte del funcionament entre clients i servidors.</i>	12
<i>Figura 4: Interaccions entre el model la vista i el controlador del patró MVC.</i>	15
<i>Figura 5: Logotips en ordre d'esquerra a dreta, AngularJS, Backbone i Ember.</i>	21
<i>Figura 6: Exemple de codi i de representació d'un SVG.</i>	23
<i>Figura 7: IDE WebStorm.</i>	24
<i>Figura 8: Repositori privat a Bitbucket.com</i>	25
<i>Figura 9: Logotips del sistema operatiu Linux i del servidor Web Apache2.</i>	26
<i>Figura 10: Diagrama de cas d'ús general.</i>	27
<i>Figura 11: Diagrama de cas d'ús d'avançar en el flux de dades.</i>	28
<i>Figura 12: Diagrama de cas d'ús de retrocedir en el flux de dades.</i>	29
<i>Figura 13: Diagrama de cas d'ús carregar fitxer de configuració RAM.</i>	30
<i>Figura 14: Diagrama de cas d'ús de editar memòria RAM.</i>	31
<i>Figura 15: Diagrama de classes general del projecte de la màquina senzilla.</i>	38
<i>Figura 16: Part del diagrama de classes, específicament la classe Ui.</i>	39
<i>Figura 17: Part del diagrama de classes, específicament la classe RAM.</i>	40
<i>Figura 18: Les altres classes que componen el projecte.</i>	41
<i>Figura 19: Estructura de directoris del projecte.</i>	42
<i>Figura 20: Fitxer de configuració de RequireJs.</i>	44
<i>Figura 21: Fitxer de configuració de RequireJs on inicialitza l'aplicació un cop carregades les llibreries i frameworks.</i>	45
<i>Figura 22: Exemple implementació a index.html del framework RequireJS.</i>	45
<i>Figura 23: Esquema de l'estructura del DOM del projecte.</i>	46
<i>Figura 24: Fragment secció "head" del index.html.</i>	47
<i>Figura 25: Representació del menú de l'aplicació en format d'escriptori.</i>	47
<i>Figura 26: Representació de l'aplicació en format mòbil.</i>	48
<i>Figura 27: Model de separació de columnes de Bootstrap.</i>	48
<i>Figura 28: Esquema del model de classes de Backbone.</i>	49
<i>Figura 29: Fitxer que representa el model del element ALU de l'aplicació.</i>	50
<i>Figura 30: Fitxer que representa la vista del element ALU de l'aplicació</i>	51
<i>Figura 31: Fitxer Javascript que representa el model la interfície UI.</i>	52
<i>Figura 32: Fitxer que representa les funcions del model de la classe UI.</i>	53
<i>Figura 33: Esdeveniments que controla la vista de la classe Ui.</i>	54

<i>Figura 34: Fitxer on es comença l'execució de tot el nucli de l'aplicació.</i>	54
<i>Figura 35: Imatge final de l'aplicació de la màquina senzilla.</i>	55
<i>Figura 36: Exemple de la visualització del bus de control i dades de l'aplicació.</i>	56
<i>Figura 37: Entrada de doble bits per a l'ALU, multiplexador i CO del IR.</i>	56
<i>Figura 38: Flux de dades i senyals de control en l'etapa de fetch.</i>	57
<i>Figura 39: Paleta de colors per seleccionar el color del flux de dades i de control.</i>	57
<i>Figura 40: Visualització de la memòria RAM en dues parts: instruccions i dades.</i>	58
<i>Figura 41: Panell d'edició manual de la RAM.</i>	59
<i>Figura 42: Consola de l'aplicació de la màquina senzilla.</i>	60
<i>Figura 43: Exemple del de "Callback" utilitzat per llegir les dades de fitxer i processar-les seqüencialment.</i>	61
<i>Figura 44: Logotip d'Amazon Web Services.</i>	65
<i>Figura 45: Panell principal d'AWS Services.</i>	66
<i>Figura 46: Panell principal del creador i configurador d'instàncies EC2 d'Amazon Web Services.</i>	66
<i>Figura 47: Panell administració d'instàncies amb detall.</i>	67
<i>Figura 48: Panell de selecció d'AMI's d'Amazon.</i>	67
<i>Figura 49: Llista dels tipus d'instàncies segons recursos hardware d'Amazon.</i>	68
<i>Figura 50: Configuració de la xarxa de la instància.</i>	68
<i>Figura 51: Configuració i selecció de particions.</i>	69
<i>Figura 52: Configuració del firewall del servidor.</i>	69
<i>Figura 53: Aplicació Putty per realitzar connexions ssh des de Windows.</i>	70
<i>Figura 54: Connexió establerta per ssh al nostre servidor d'Amazon.</i>	71
<i>Figura 55: Trasnferència del projecte al servidor via Filezilla.</i>	72
<i>Figura 56: Contacte i llista de reports de cada versió de l'aplicació.</i>	73
<i>Figura 57: Resum dels botons que fa servir l'aplicació de la Màquina Senzilla.</i>	98

14.2 Llicència sobre la qual està definit el projecte

GNU General Public License v3

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface

presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is

effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any

other way, but it does not invalidate such permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or

- (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
 - d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
 - e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user,

“normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an

implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently,

if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives

whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To

“grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the

covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version”

applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD

PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

15. Manual d'usuari

En aquest apartat explicarem els requisits tècnics que ha d'utilitzar l'usuari i quines son les seves opcions que te per utilitzar l'aplicació.

15.1 Requisits tècnics

15.1.1 Compatibilitats amb el navegador

Els navegadors on s'han fet proves de visualització han sigut: Google Chrome, Mozilla Firefox, Opera, Internet Edge, Safari i Internet Explorer. L'únic navegador que dona problemes a l'hora de visualitzar les imatges SVG i les seves animacions és Internet Explorer.

15.1.2 Resolució del navegador

Respecte a la resolució mínima per a una correcta visualització s'ha optat per la 1024x768. Si la resolució es més baixa es pot trobar que la visualització de l'aplicació no sigui del 100% correcta.

15.2 Funcionament de l'aplicació

15.2.1 Botons de control de l'aplicació

Clock -: Retrocedeix el flux de dades fins al flanc de rellotge anterior. Ho pot fer fins a l'estat inicial del programa.

Clock +: Simula el flanc de pujada de la màquina senzilla. Aquest botó fa que per pantalla es visualitzi el flux de dades.

Reset: Aquest botó reinicia la màquina però no la memòria.

Kill: Inicialitza tota la màquina senzilla, incloent-hi la memòria.

Selecionar fitxer: Ens permet carregar un fitxer amb extensió ".txt" amb instruccions i dades per carregar a la memòria.

Editar memòria: Ens permet editar la memòria directament.

Llegenda de colors: Botons que ens permeten canviar els colors dels senyals de control i del bus de dades.

Assemblador: CheckBox que ens permet visualitzar en mode assemblador el contingut de la màquina.

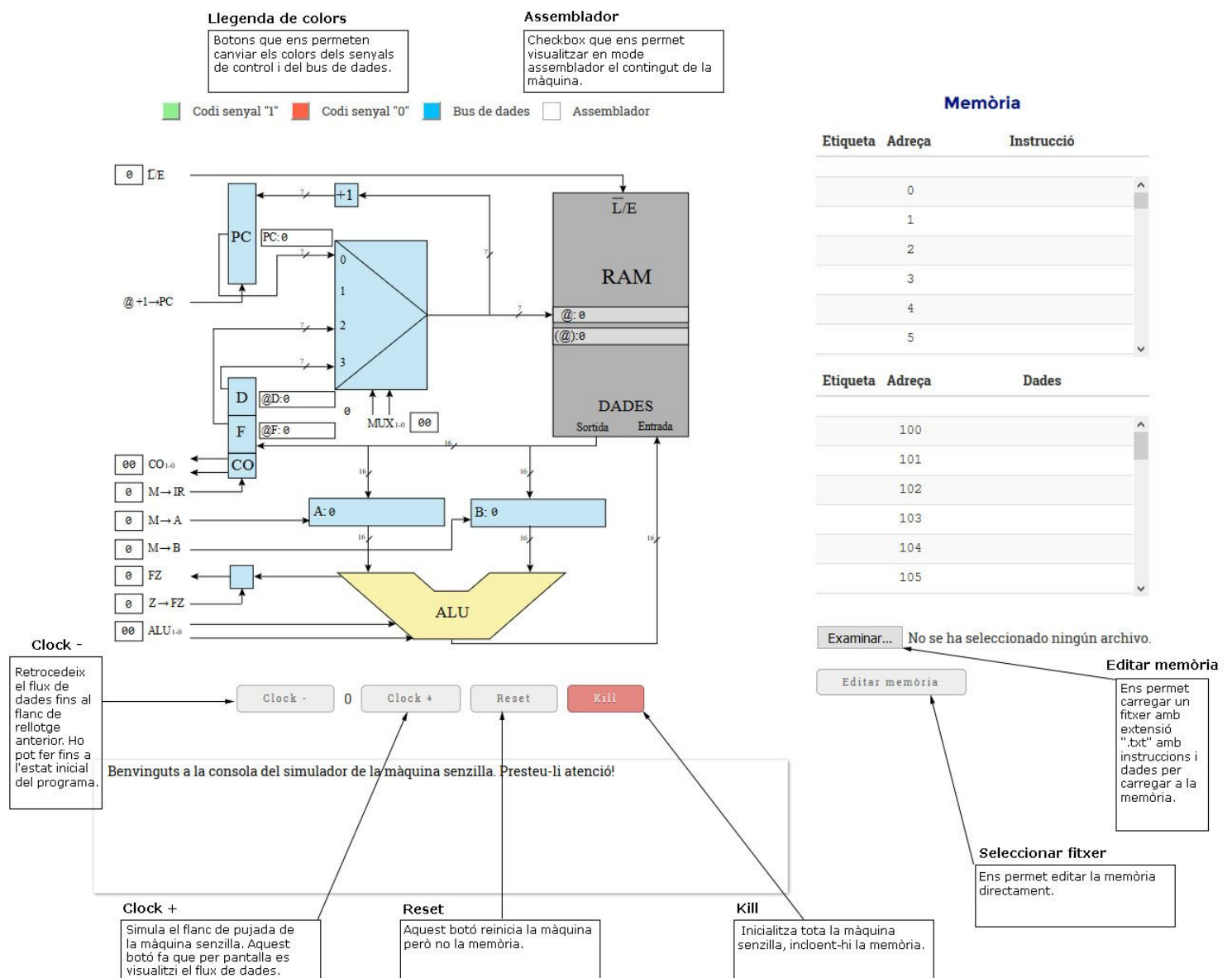


Figura 57: Resum dels botons que fa servir l'aplicació de la Màquina Senzilla.

15.2.2 Càrrega del fitxer de configuració i edició de la memòria

Per a l'entrada d'instruccions la nova aplicació de la màquina senzilla processarà línia per línia les instruccions entrants que tindran una sintaxi específica:

- Etiqueta o numero (Opcional): Serà un string seguit de dos punts ":" (ex: "inici:"). També podrà ser un enter seguit amb dos punts, en aquest cas ens guardarà la instrucció en la posició d'aquest enter. (ex: "15:").
- Sense el marcador s'aniran omplint automàticament els espais buits per a instruccions lliures.
- Tipus operació: Les normals de la màquina senzilla, (add | mov | cmp | beq).
- Operador Font (no necessari en "beq"): Pot ser un enter que apunta a una direcció de memòria o un string que apunta a una etiqueta. En el cas del "beq" es pot ignorar. Si hi és ha de estar separat sempre amb la coma "," per després definir l'operador destí. En base decimal
- Operador Destí: Igual que l'operador font pot ser un enter or un string, però aquest és sempre obligatori. En base decimal
- El punt i coma ";" (Opcional): Serveix per finalitzar la instrucció però es també pot ser opcional.

Exemples:

#Escriu amb etiqueta, a la primera posició lliure per instruccions (add 100, 111)
inici: add 100, 111

#Escriu a la posició 5 de memòria la instrucció (cmp 110, 110)
5: cmp 110, 110;

#Escriu a la primera posició lliure I resol l'adreça de la etiqueta
add 100, un

beq inici;

#Interpretat serà igual que final: add 0 0.
final:

Per a l'entrada de dades la sintaxi és més senzilla.

- Etiqueta o numero (Opcional): Serà un string seguit de dos punts ":" (ex: "zero:"). També podrà ser un enter seguit amb dos punts, en aquest cas ens guardarà el valor en la posició d'aquest enter. (ex: "100:").
-
- Sense el marcador s'aniran omplint automàticament els espais buits per a les dades.
- Valor: El valor de la dada en base decimal.

Exemples:

245

zero: 0

110: 4

Finalment hi ha dos maneres de carregar el programa, des de un fitxer de text o editant directament amb el botó d'edició de memòria.

