

# Creació d'una nova plataforma i procés de desenvolupament

## Student Groups

### Inici de Sessió

Nom d'Usuari:

Contrassenya:

Entra

Marc Clascà Ramírez

Tutora: Anna Basart

Desembre 2015



# Índex

1. Introducció	5
2. Anàlisi de la necessitat	8
2.1. Valoració dels usuaris	9
2.2. Plantejament d'una nova plataforma	10
3. La idea de l'aplicació: Student Groups	12
3.1. Funcions	12
3.2. Estructura usuaris i grups	12
4. Tria de les eines	13
4.1. Backend: Parse.com	13
4.2. Frontend: AngularJS	14
4.3. Altres eines pel desenvolupament	14
5. Formació, Recerca d'Informació i Solució de Problemes	16
5.1. Curs de Codecademy	16
5.2. Documentació i Recursos per Internet	17
5.3. La Solució de Problemes Concrets	17
6. Arquitectura de l'aplicació	18
6.1. Característiques de AngularJS	18
6.1.1. Disseny Model—View—Controller	
6.1.2. Aplicacions d'una Sola Pàgina	
6.1.3. Two way data-binding	
6.1.4. Injecció de dependències	
6.2. Separació per mòduls	19
7. Desplegament	24
7.1. Fases de desenvolupament i servidors	24
7.2. Accés a l'aplicació i el codi	24
8. Conclusions	26
9. Bibliografia	27
Annexos	29



# 1. Introducció

El curs 2010-2011 el govern de la Generalitat va començar a 369 centres d'educació secundària públics i a 244 de concertats el programa *Educat 1x1* amb l'objectiu de digitalitzar l'educació. Aquest nou projecte implicaria que cada alumne tingués un ordinador portàtil amb els llibres digitals i que les classes es desenvolupessin d'una manera totalment nova fins aleshores, amb nous equips com les pissarres digitals. Però hi ha una altra part d'aquesta *digitalització* que no està a les aules i que també implica una gran part dels canvis: parlo de les plataformes i programes de gestió acadèmica, control d'assistència, infraestructura de xarxa, comunicació... Aquests programes estan en el dia a dia a l'institut i per aquest motiu han d'estar fets a mida per als professors i estudiants.

Aquest projecte va començar el mateix any que jo començava l'ESO i des de primer fins a quart he pogut veure com evolucionaven totes aquestes eines i com s'anaven solucionant els problemes mentre se'n creaven de nous. Poc a poc aquesta integració s'anava fent més gran, ampliant-la amb noves funcions com l'entrega de treballs per internet, el control d'assistència digital i pagaments a través d'un TPV<sup>1</sup>.

Durant aquest temps he sentit la necessitat d'investigar i qüestionar com estava muntat tot aquest nou sistema i, fins i tot, de posar-lo a prova. Un exemple és el solucionari de les activitats dels llibres digitals de La Galera Text que vaig fer durant 3r i 4t. Aquest programa estava basat en un error d'autenticació dels llibres digitals que permetia accedir als arxius PDFs amb les solucions escrites sense haver de tenir un compte de professor. Volia descobrir els engranatges de tot aquest sistema, d'una banda pel fet que mai havia pogut veure una infraestructura de TIC tan àmplia, i de l'altra perquè que m'implicava directament a mi al ser alumne i per tant volia poder mostrar què en pensava. Un projecte que implica tants canvis en el sistema educatiu hauria de tenir en compte tot el seu col·lectiu a l'hora d'aplicar-se.

Vaig començar a interessar-me pels ordinadors quan tenia onze anys, quan ja descarregava i instal·lava tot tipus de programes i "jugava" amb les configuracions del sistema operatiu. Les meves primeres línies de codi van ser en HTML, que van donar lloc al meu primer projecte, que va ser una pàgina web que em vaig posar d'inici al navegador de l'ordinador de l'institut i on tenia un horari i els enllaços als llocs web que més fèiem servir (Moodle, Àtria, La Galera Text...). A partir d'aquí vaig anar cap als llenguatges de programació per pàgines web com *Javascript* o *PHP*. Al cap d'un temps vaig fer el salt a les aplicacions mòbils per *iOS* utilitzant els llenguatges *Objective-C* i *Swift*. A partir d'aquest punt el meu interès per la programació en general va créixer de manera exponencial, cosa que em va portar a implicar-me més en la comunitat de desenvolupadors. Volia

---

<sup>1</sup> Terminal Punt de Venda

portar tot allò que havia treballat des de casa al món real, per començar creant projectes reals (com la pàgina web [afinacions.cat](http://afinacions.cat) i l'aplicació Pianoconcert App) i també assistint a esdeveniments per a la comunitat.

El Juny de 2015 vaig tenir l'oportunitat d'assistir a la WWDC<sup>2</sup> d'Apple, una conferència destinada als desenvolupadors d'aplicacions per a les seves plataformes on es realitzen una sèrie de tallers i ponències presentades pels mateixos enginyers de l'empresa que s'encarreguen de desenvolupar els dispositius i sistemes operatius. Aquest va ser el meu primer contacte real tant amb gent que es dedica professionalment al desenvolupament de programari com amb altres estudiants de la meua edat que també hi estan interessats. Vaig veure infinitats de projectes enginyosos, vaig escoltar i valorar idees i fins i tot en vaig tenir de pròpies. Va ser una setmana d'investigació i aprenentatge en un ambient creatiu que me va portar a voler aportar les meves solucions al món i, doncs, fer més projectes reals. Una de les influències que ha tingut el meu Treball de Recerca és una noia dels EUA, que durant aquells dies em va explicar la seva idea d'unes eines tecnològiques per a l'educació que fossin adequades i fàcils d'utilitzar. Ella i un company seu estaven desenvolupant en aquells moments una aplicació per a dispositius mòbils per al seu institut tant per a la gestió interna del centre com per a alumnes i famílies.

Així doncs, de la llista de projectes que se'm van acudir vaig acabar triant per a fer com a Treball de Recerca el programa de comunicació per a instituts. A partir d'aquí, i tenint en compte tot el que he explicat en aquesta introducció, sorgeix el primer objectiu d'aquest treball:

- **Creació d'una nova eina tecnològica acadèmica que sigui fàcil d'utilitzar encara que faci poques coses.**

Un cop plantejats el treball de recerca i el primer objectiu, vaig pensar en més objectius que m'agradaria assolir fent aquest treball. Tot i que he desenvolupat molts projectes i he provat moltes eines, la meua experiència es basa només en això, la prova i l'error. Mai he realitzat un desenvolupament d'una aplicació o programa com si fos un producte real, és a dir, amb els seus usuaris, el seu públic i una funció clara. Així doncs, aquest és el meu segon objectiu:

- **Fer el procés de desenvolupament d'un producte de software real.**

Finalment, un cop vaig haver decidit com seria el meu treball de recerca i quin seria el projecte vaig voler posar-me un últim objectiu més tècnic. Volia desenvolupar l'aplicació utilitzant unes plataformes i *frameworks* que no hagués utilitzat mai per així aprendre a fer-los servir. Per tant, l'últim objectiu que plantejo és:

- **Aprendre a desenvolupar aplicacions web utilitzant AngularJS i [Parse.com](http://Parse.com) a partir de la realització d'un projecte.**

---

<sup>2</sup> Conferència Mundial de Desenvolupadors, de les sigles en anglès *WorldWide Developers Conference*, veieu <http://apple.com/wwdc>

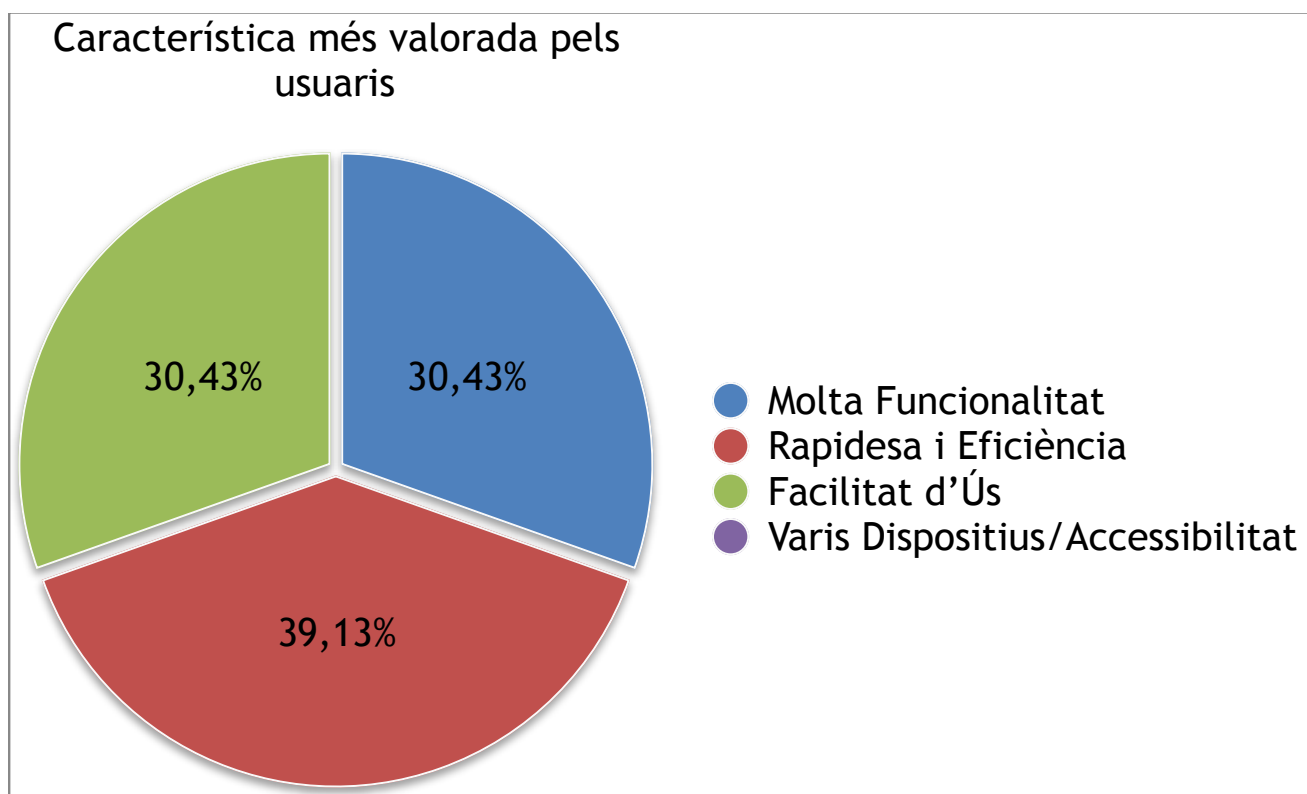
El guió que he seguit en aquest treball escrit serà el següent: Una primera part en què analitzaré el problema i buscaré una solució, descrivint la meva aplicació; i una segona part de caire més tècnic en què parlaré, primer, de les eines i els recursos que he fet servir i com he après a fer-los servir i, segon, sobre la l'arquitectura de desenvolupament que he aplicat. La major part del treball, però, quedarà reflectida en el desenvolupament de l'aplicació.

## 2. Anàlisi de la necessitat

Davant de la gran quantitat d'eines i plataformes que ja hi ha i que ja es fan servir en l'àmbit de l'educació, la meua aplicació necessita uns punts forts que la facin destacar per sobre les altres i que eviti que no es converteixi en una aplicació més que només dificulta la tasca de d'educar. Així doncs, les dues característiques que he tingut en compte alhora de realitzar el treball i són:

- Que l'**adequació** i la **funcionalitat** siguin bones i clares i que la relació entre aquestes dues característiques estigui equilibrada.
- Que l'aplicació estigui pensada i feta per als usuaris que la faran servir, és a dir, per els docents, alumnes i les famílies d'aquests.

A continuació explicaré els dos atributs als quals faig referència al primer punt alhora de valorar una eina TIC i com haurien de ser perquè es compleixi la necessitat exposada. Quan parlem d'adequació ens referim a l'adequació al context d'utilització d'aquesta eina, és a dir, si les



funcions que té i la manera com està desenvolupada, independentment de si funciona bé, són correctes per a la funció que es donarà a aquesta eina. Per exemple, imaginem un context en què necessites un programa per administrar els estocs d'una petita botiga. Paguem un desenvolupador perquè ens faci un programa i resulta que aquest programa té moltes funcions i característiques diferents. En el nostre cas, el més probable és que no ens acabi funcionant bé ja que la tasca que nosaltres volíem solucionar era molt simple i clara (mantenir un registre dels



estocs) i la majoria de la funcionalitat que porta el programa és innecessària i només el farà més lent i difícil d'utilitzar.

El segon atribut, la funcionalitat, fa referència a les diferents coses que fa l'aplicació, és a dir, les funcions que té i el que això ens aporta. La rapidesa i eficàcia de l'aplicació i la facilitat d'ús dependran directament de com i quina sigui aquesta funcionalitat. Per això, perquè una aplicació funcioni bé, ha de tenir poques funcions però que aquestes siguin ràpides i desenvolupin la seva tasca de forma eficaç.

Així doncs necessitem una aplicació que solucioni un problema en concret i per tant les funcions que tingui estiguin programades per complir aquest objectiu de manera eficaç, sense fer l'aplicació més lenta i complicada d'utilitzar. L'equilibri que busquem és poca funcionalitat però dedicada exclusivament a solucionar la necessitat.

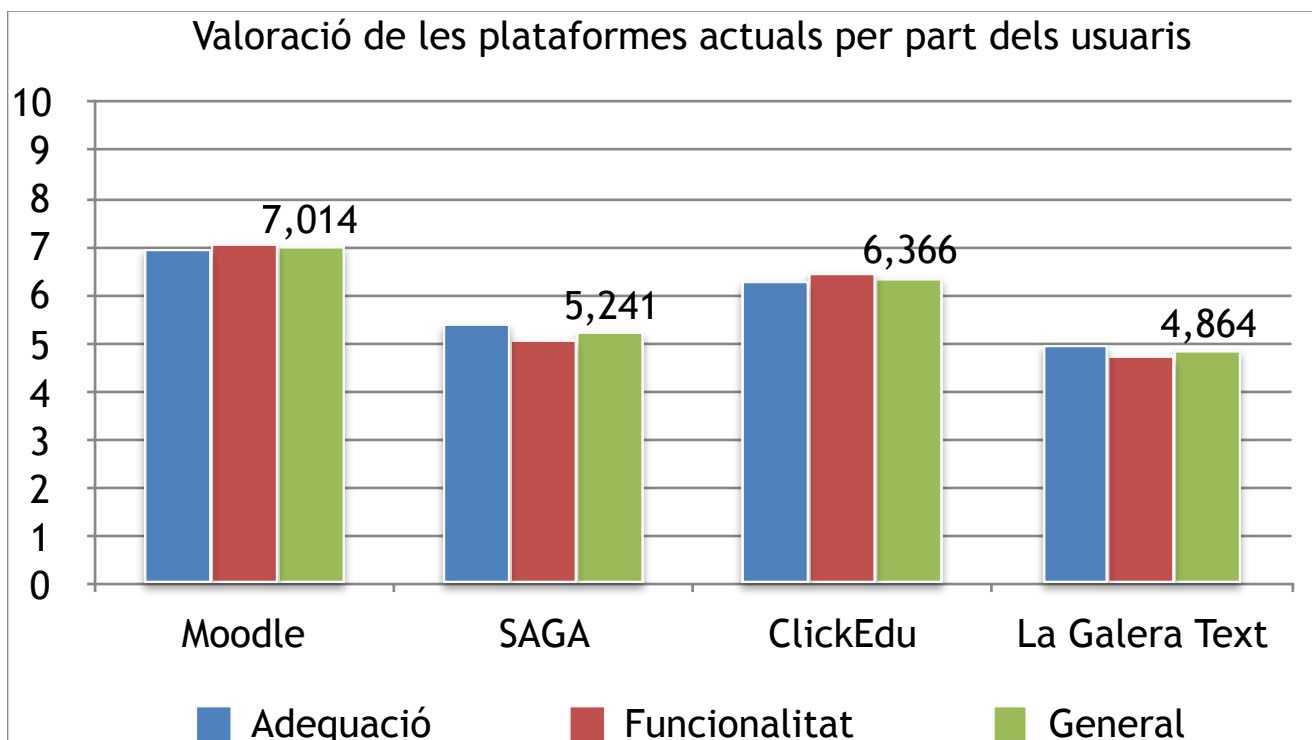
## 2.1. Valoració dels usuaris

Per definir aquesta aplicació (tal i com faig referència al segon punt) cal tenir en compte les opinions i necessitats dels usuaris finals que la faran servir, en aquest cas els professors i els alumnes. Per això he decidit realitzar una enquesta en què, en primer lloc, demano l'opinió sobre les plataformes i eines que ja es fan servir a l'institut (v.gr. Moodle, ClickEdu, SAGA...) i, en segon lloc, pregunto com hauria de ser la nova aplicació que plantejo en aquest treball. Tot seguit faré un anàlisi dels resultats i als annexos podreu trobar l'enquesta amb cada resposta.

Per començar vaig voler saber quina és l'opinió general que es té del projecte Educat 1x1 i del que això ha comportat. A la pregunta *Com valoraries la integració que es fa actualment de la tecnologia en l'educació?*, en què s'havia de escollir un número entre l'1 i el 10, la mitjana ha estat de 5,8. Abans d'endinsar-nos a valorar les diferents plataformes que fem servir actualment per separat pensem una mica en aquest número. Podríem dir que la satisfacció general a l'Institut de La Roca sobre el projecte Educat 1x1 té un "aprobat justet". Per ser un projecte que condiciona el dia a dia a en l'educació penso que hauria de tenir un grau de satisfacció més elevat entre els usuaris..

Aquesta és la valoració de les eines que utilitzem actualment a l'institut: Moodle, ClickEdu, els llibres digitals de La Galera Text i el programa de gestió acadèmica de la Generalitat, el SAGA (Sistema d'Administració i Gestió Acadèmica). A la meua enquesta demano la valoració de cada una d'aquestes plataformes segons els dos atributs dels quals he parlat abans en aquest apartat, la funcionalitat i l'adequació. Podem veure aquests valors representats al següent gràfic:

La tercera columna mostra una valoració general, que és la mitjana entre les dues valoracions anteriors. Podem veure que la plataforma més ven valorada, amb un 7, és el Moodle. El Moodle s'utilitza des de fa molt temps (abans de l'inici del projecte), això vol dir que hi ha hagut bastanta formació i molta experiència d'ús. Això ens fa pensar que gràcies a que gairebé tothom ja el sap fer servir el programa rep unes millors valoracions, cosa que ens diu que el principal problema que



tenen aquestes plataformes és les dificultats d'ús en què es troben els usuaris el primer dia que es posen davant la pantalla per a fer-lo servir. Una plataforma acadèmica hauria de poder-se fer servir fàcilment des del primer dia sense necessitat de formació específica.

Una segona conclusió que podem treure és que la valoració relativament dolenta que rep el SAGA pot ser deguda a que és una eina d'ús imposat pel sistema educatiu. Si el programa ja no és molt bo i a sobre l'has de fer servir obligatòriament això fa que encara t'agradi menys. La resta de valoracions són un 6,4 per al ClickEdu, una valoració relativament bona tot i ser una plataforma que s'ha començat a fer servir fa poc, potser degut a que té una interfície més amable; i un suspès per a La Galera Text, que rep unes valoracions dolentes tot i que actualment (però des de fa poc) ja ha passat a utilitzar-se com a única editorial de llibres digitals.

## 2.2. Plantejament d'una nova plataforma

Primer de tot vaig voler saber quina és la característica que més valoren els usuaris d'un programa informàtic qualsevol. Per això els vaig fer escollir entre les opcions Molta Funcionalitat, Rapidesa i Eficiència, Facilitat d'Ús i Varis Dispositius/Accessibilitat; i els resultats són els següents:

La majoria dels usuaris prefereixen una aplicació ràpida i que funcioni bé.

Acabem l'enquesta amb les preguntes sobre idees per a la nova aplicació. De tots els comentaris rebuts, he sintetitzat una llista de característiques demandades pels usuaris:

- Control d'assistència
- Missatgeria
- Control i avaluació de tasques

- Resultats
- Disponibilitat per a tots els dispositius
- Comunicació àgil a través de correu electrònic.
- Registre de disciplina

Amb totes aquestes variables analitzades podem plantejar una nova plataforma acadèmica per ús a l'institut que tingui una bona rebuda a la comunitat educativa i que no es converteixi en un altre maldecap.

### 3. La idea de l'aplicació: *Student Groups*

L'aplicació és una plataforma de comunicació acadèmica, és a dir, l'objectiu per a la qual està pensada és la interacció entre professors, alumnes i famílies. Així doncs incorporarà funcionalitats tals com la gestió d'alumnes i grups per part dels professors, l'assignació de deures i la consulta d'aquests per part dels alumnes i els seus pares o la suggerència d'entrevistes amb els familiars.

Per assegurar que l'aplicació sigui accessible des de tots els dispositius s'hi accedirà des d'una pàgina web, per tant serà una aplicació web.

#### 3.1. Funcions

Les funcions bàsiques que he decidit implementar són:

- a) Administració de tasques i deures: els professors poden assignar tasques als seus grups o a algun alumne en concret. Els alumnes que tinguin alguna tasca assignada ho podran veure a la seva pàgina d'inici i a la pàgina de tasques.
- b) Resultats: Els professors podran avaluar i posar notes als alumnes i aquests les poden consultar.
- c) Assignatures: Cada alumne estarà dintre de diversos grups. Els grups equivalen a una assignatura, i cada grup té un responsable, el professor. Els grups tenen un horari, l'horari d'aquella assignatura. El professor pot consultar els grups dels quals ell n'és professor i veure els alumnes.
- d) Comunicació amb els pares: a través de la pàgina de l'alumne un professor pot enviar un missatge als seus pares directament.

Aquestes són les funcions que crec que sóc capaç implementar inicialment, però poc a poc es pot anar ampliant el programa per afegir-hi totes les altres característiques.

A l'annex 3 es pot veure un disseny inicial fet a mà de la pàgina inicial de l'aplicació, que resumeix les funcions de les que he parlat abans.

#### 3.2. Estructura usuaris i grups

Cada alumne i professor tindrà un compte en aquesta plataforma. Un grup, un curs i una assignatura són equivalents.

El professor crea la seva assignatura, per exemple *Llengua Catalana 2n C ESO*, afegeix els alumnes que fa aquella assignatura i designa l'horari d'aquella assignatura, els dies de la setmana i les hores que es fa.

L'alumne forma part de diversos grups, tants grups com assignatures faci. Pot consultar la informació específica de cada assignatura.

## 4. Tria de les eines

El primer pas per desenvolupar l'aplicació és triar les eines i els *frameworks* amb els quals la voldré desenvolupar, i a continuació rebre la formació necessària per aprendre a fer-los servir. Podem separar una aplicació en dos grans parts:

- a) *Backend*, o part del servidor, és la part que conté tota la lògica del programa i que s'encarrega d'emmagatzemar i administrar les dades. Conté bases de dades, control d'accés dels usuaris, i està en servidors a internet. Aquesta part és intangible, és a dir, l'usuari no la pot veure. Només la part del client hi pot accedir, a través d'una adreça d'internet.
- b) *Frontend*, o part del client, és la part que conté la interfície gràfica i la part amb què l'usuari interacciona. S'encarrega de mostrar les dades i conté les eines per veure i modificar les dades. Es comunica amb la part del servidor a través d'internet. La part del client pot tenir diverses formes, com pot ser una pàgina web, una aplicació per a un dispositiu mòbil o un programa d'ordinador.

### 4.1. *Backend*: Parse.com



Per al desenvolupament del *Backend* he escollit una plataforma anomenada **Parse**. Es tracta d'un servei de l'empresa Facebook que conté les bases de dades preparades i la lògica de servidor preparada perquè el desenvolupador es pugui dedicar a fer l'aplicació en sí i s'estalviï tota la feina que comporta posar en marxa un servidor, una feina que és pràcticament idèntica per a qualsevol aplicació. He triat aquesta opció per aquest estalvi de feina que he dit i també perquè és una plataforma nova i que s'adapta a les últimes tecnologies. També perquè els límits d'ús que té la versió gratuïta són molt elevats, pràcticament impossibles de sobrepassar si no tens una gran quantitat d'usuaris. És una plataforma que no havia fet servir mai anteriorment en cap dels meus projectes.

Un dels avantatges que té Parse és que disposa de SDK nadius per a molts llenguatges de programació. D'aquesta manera és molt fàcil connectar qualsevol client, independentment de com estigui programat. En el meu cas descarregaré i faré servir el SDK de Javascript perquè

programaré el client en aquest llenguatge. Un SDK (*Software Development Kit*) és un conjunt d'eines i funcions de desenvolupament de software que permeten al programador crear aplicacions per un sistema en concret, en aquest cas la plataforma Parse.

El fet que hagi escollit una plataforma de *Backend* en comptes de decidir programar la part del servidor jo mateix també és perquè així em puc concentrar més en l'aplicació i les seves funcions i oblidar-me de tota la lògica igual per a totes les aplicacions.

## 4.2. Frontend: AngularJS



Per al desenvolupament del *Frontend* també he triat una plataforma bastant nova i que no he fet servir mai. Es diu **AngularJS**, és un framework MVW en llenguatge de programació Javascript per escriure aplicacions web i està fet i mantingut per l'empresa Google. MVW significa, en anglès, *Model — View — Whatever*, i en català Model — Vista — “El que sigui”. És una arquitectura de desenvolupament que et dóna flexibilitat per treballar amb qualsevol de les altres arquitectures, MVC o MVVM. Més endavant parlaré sobre aquests conceptes.

En resum, AngularJS converteix el llenguatge HTML de disseny de pàgines web en un llenguatge molt més dinàmic, cosa que el fa més útil i fàcil de fer servir alhora de programar aplicacions web que administrin tantes dades com la que plantejo. Això et permet concentrar-te en les dades que has de mostrar i com les has de mostrar per aconseguir una interfície intuïtiva per l'usuari, sense haver de pensar en la lògica que això necessita.

He triat aquest framework perquè utilitza les últimes tecnologies web, és fàcil d'utilitzar i ràpid per l'usuari. També perquè no l'havia utilitzat mai i aquesta és una forma de que aprengui a fer-lo servir, ja que cada vegada s'utilitza més en projectes reals.

## 4.3. Altres eines pel desenvolupament

L'ús d'eines i utilitats no s'acaba aquí. La comunitat de desenvolupadors s'encarrega de fer programes de codi obert que realitzen tasques que faciliten molt la feina alhora de desenvolupar altres programes. Sense aquestes eines, utilitats o llibreries desenvolupar software seria molt més feixuc i costós, i es perdria molt temps en tasques repetitives. Aquestes són els eines que jo he fet servir en el meu desenvolupament:

- a) **Sublime Text 2**, és l'editor de text de codi que he escollit per escriure l'aplicació. És ràpid i presenta una interfície senzilla i potent, que evita qualsevol tipus de distracció. És un software

de codi obert, cosa que permet l'extensió de la seva funcionalitat gràcies a plugins, petits trossos de codi que s'incrusten al programa per afegir-li una funció en concret. És gratuït i multi-plataforma.



b) **Bower**, és un administrador de paquets per a pàgines web. Permet descarregar i afegir llibreries frontend al teu projecte molt fàcilment, amb un simple comandament a la consola. Una llibreria és un conjunt de funcions preparades per utilitzar al teu projecte.



c) **Grunt**, és un *task-runner*, és a dir, un programa encarregat de realitzar unes tasques o processos repetitius automàticament. Simplement el configures abans amb les tasques que vols que realitzi i quan el necessites l'actives a través d'un comandament a la consola. Realitza tasques repetitives del desenvolupament com compilar l'aplicació, reduir els arxius, engegar un servidor web per veure l'aplicació en funcionament o realitzar unes proves al codi.



d) **Yeoman**, és una eina que permet construir els fonaments d'una aplicació. Quan l'actives et pregunta els frameworks que voldràs utilitzar per programar la teva aplicació, les llibreries que vols integrar i les tasques que vols realitzar. Gràcies a la seva integració amb les dos eines anteriors (Bower i Grunt) realitza tots aquests processos en un moment i et lliura els fonaments o la "façana" (d'aquí el nom *scaffolding tool*) de la teva aplicació, preparada per començar a desenvolupar.



```
Web App — bash — ttys000

app/scripts/services/userservice.js
  line 13 col 27 Missing semicolon.
  line 46 col 11 Expected an assignment or function call and instead saw an expression.

  17 warnings

Warning: Task "jshint:all" failed. Use --force to continue.

Aborted due to warnings.

Execution Time (2015-12-05 18:33:36 UTC)
loading tasks          7ms    2%
loading grunt-newer    14ms    3%
newer:jshint:all        14ms    3%
loading grunt-contrib-jshint 118ms 28%
jshint:all              262ms 63%
Total 416ms

Completed in 0.756s at Sat Dec 05 2015 19:33:37 GMT+0100 (CET) - Waiting...
>> File "app/partials" added.
Completed in 0.000s at Sat Dec 05 2015 19:35:10 GMT+0100 (CET) - Waiting...
>> File "app/partials/login.html" added.
Completed in 0.000s at Sat Dec 05 2015 19:35:32 GMT+0100 (CET) - Waiting...
^C

Execution Time (2015-12-05 16:36:29 UTC)
watch                  7h 21m 1.3s 100%
Total 7h 21m 6.7s

MacBook-Pro-de-Marc:Web App marcclasca$
```

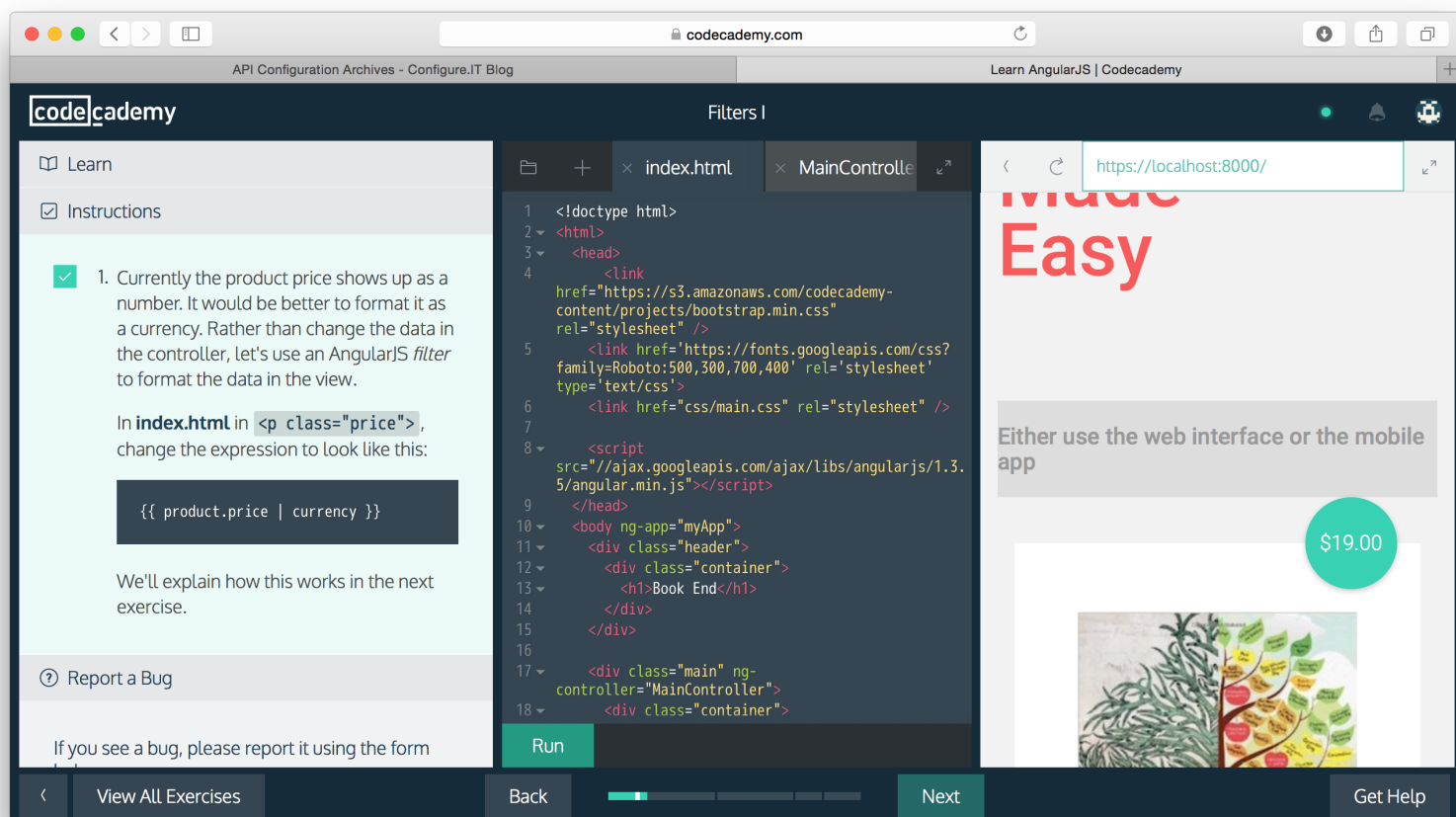
## 5. Formació, Recerca d'Informació i Solució de Problemes

Des de que vaig començar a programar la meua formació en aquest camp ha estat totalment autodidacta. Des del primer dia aquesta formació s'ha basat en recursos gratuïts d'internet, vídeo tutorials i articles, majoritàriament en anglès. L'aprenentatge és diari, a base de realitzar projectes i exemples reals i de solucionar un a un cada problema amb el que em vaig trobant. Això implica, doncs, que la realització d'aquest treball de recerca no deixa de ser una fase en el meu aprenentatge. Com qualsevol vegada que començo a utilitzar una nova eina, començo amb algun curs guiat introductori sobre aquesta eina, que continuo al llarg del desenvolupament amb la consulta de la documentació i la recerca d'articles i complemento amb el sempre fidel *Stackoverflow*.

### 5.1. Curs de Codecademy

Hi ha varies pàgines web que ofereixen cursos guiats gratuïtament, o almenys una part dels cursos. Aquestes són [codeschool.com](https://www.codeschool.com), [codecademy.com](https://www.codecademy.com) o [khanacademy.org](https://www.khanacademy.org), de les quals jo he tret bastanta informació. Per començar amb AngularJS jo vaig escollir el curs de Codecademy: <https://www.codecademy.com/learn/learn-angularjs>

La següent imatge és una captura de pantalla del curs, quan el vaig realitzar a través d'una pàgina web:





## 5.2. Documentació i Recursos per Internet

La part més important de la formació és la recerca d'informació per internet durant el desenvolupament. Per cada dificultat que em trobo alhora d'escriure alguna funció o característica del programa, busco a internet com solucionar-ho. Per fer això hi ha la **documentació**. Són documents didàctics que t'ensenyen a fer servir les llibreries i els frameworks. Cada un d'aquests sistemes porta la seva documentació, escrita per l'organització que a creat el sistema, i els desenvolupadors la consulten contínuament per saber com implementar una característica del sistema en qüestió. Així doncs, el framework AngularJS que faig servir té la seva documentació oficial: <https://docs.angularjs.org/>; igual que Parse, que també té la seva documentació en diversos llenguatges de programació, en el meu cas la documentació en Javascript: <https://parse.com/docs/js/guide>.

Les documentacions de software normalment tenen un índex, en què hi ha la informació ordenada o bé segons les funcions o propietats que vulguis fer servir, o bé segons la realització de tasques en concret, per exemple, com inicialitzar i configurar l'aplicació.

## 5.3. La Solució de Problemes Concrets

Tota la resta de problemes amb els que em trobo més enllà dels mètodes que ja he dit només tenen dos solucions: o bé fer més recerca per internet i intentar trobar un article on parli d'un aspecte relacionat al meu problema (bastant improbable i ineficax), o bé utilitzar una pàgina de preguntes i respostes com **StackOverflow**. StackOverflow és una pàgina web en que els desenvolupadors poden penjar qualsevol pregunta relacionada amb qualsevol aspecte de la programació de software. Aquestes preguntes són respostes per membres de la mateixa comunitat de desenvolupadors. Per tant, tu pots preguntar i respondre alhora.

És una pàgina àmpliament utilitzada i coneguda entre la comunitat, el que la fa molt efectiva. Normalment tens una resposta al teu problema el mateix dia de penjar la pregunta. El que la fa més potent, però, és el fet que totes les respostes queden guardades i són accessibles públicament, de manera que si hi ha algú que ja s'ha trobat amb el mateix problema que tu i ho ha preguntat tu podràs consultar les respostes i resoldre el teu problema sense necessitat de penjar una altra pregunta. La gran quantitat de moderadors i administradors la fan una pàgina molt eficaç i una font d'informació molt veraç.



## 6. Arquitectura de l'aplicació

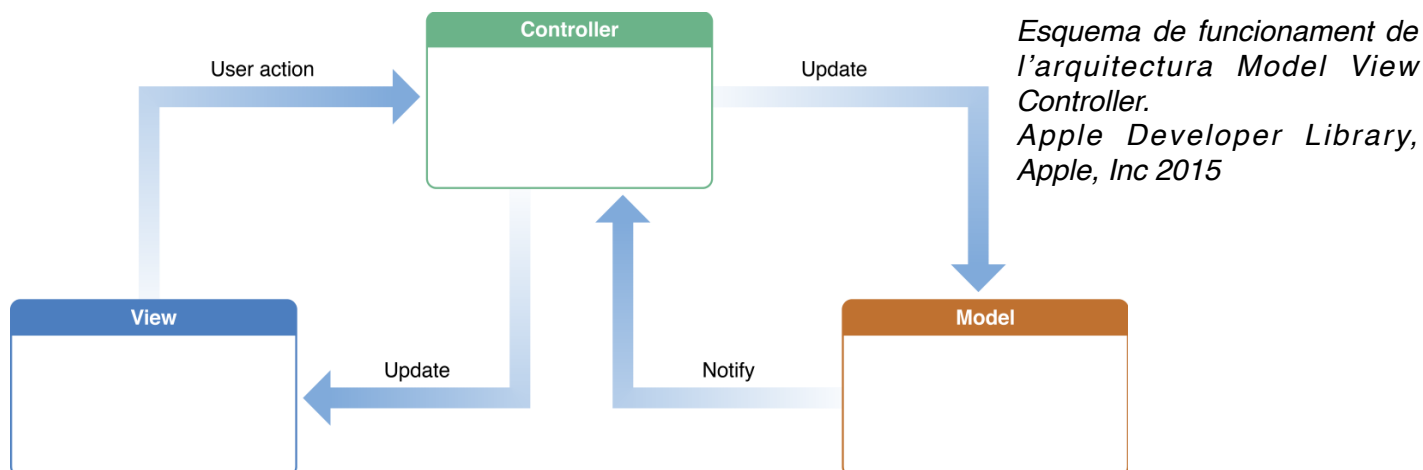
### 6.1. Característiques de AngularJS

#### 6.1.1. Disseny Model—View—Controller

AngularJS té un disseny arquitectural de MVW (Model—View—Whatever), per aquesta raó implementaré l'arquitectura MVC o Model—View—Controller.

Aquesta arquitectura defineix 3 parts en l'aplicació, interconnectades entre elles, separant d'aquesta manera la vista de les dades i la lògica

- a) El **model**: Conté i gestiona la informació. Té les funcions necessàries per emmagatzemar, modificar i donar i revocar l'accés a aquesta informació
- b) La **vista**: és la presentació visual de les dades, la interfície gràfica amb la qual l'usuari interacciona.
- c) El **controlador**: s'encarrega d'intervenir entre el model i la vista i notifica els canvis. Està atent a les interaccions que l'usuari fa a la vista i les comunica al model per canviar les dades si és necessari. També fa la funció inversa, si el model canvia les dades, el controlador s'encarrega de notificar els canvis a la vista perquè el canvi es vegi reflectit a la pantalla de l'usuari.



#### 6.1.2. Aplicacions d'una Sola Pàgina

El disseny MVC permet tenir aplicacions web d'una sola pàgina, és a dir, no cal refrescar la pàgina i canviar l'adreça al navegador per canviar les vistes: el controlador s'encarrega de fer aquesta feina. Les comunicacions amb el servidor es realitzen en moments clau sense que l'usuari se n'adoni. Tot i que la pàgina no canviï mai, les noves tecnologies web permeten que l'usuari encara tingui la sensació que les pàgines van canviant. La lògica, per això, es manté en

una sola pàgina. El principal avantatge d'aquesta característica és el fet que el navegador controla l'estat de l'aplicació tota l'estona, i no depèn de les respostes d'un servidor.

---

### 6.1.3. *Two way data-binding*

Una altra característica és la renderització de dades bidireccional. Aquesta característica permet la sincronització del model i la vista, de manera que els canvis que hi ha a la vista estan immediatament reflectits a les dades del model, i els canvis de dades es veuen immediatament reflectits a la interfície de l'usuari. La renderització d'aquestes dades a la pàgina web es realitza utilitzant els caràcters `{{ i }}` al codi HTML.

---

### 6.1.4. Injecció de dependències

La injecció de dependències és la característica que s'encarrega de proveir-nos amb els components de software que necessitem. La injecció consisteix en passar la dependència (una llibreria o un objecte) a un altre objecte dependent. Les dependències necessàries s'escriuen com a arguments a la funció de cada mòdul, entre els caràcters `[ i ]`.

## 6.2. Separació per mòduls

En una aplicació feta amb AngularJS cada part de l'aplicació és un mòdul i la combinació entre tots fa el programa sencer. Aquest esquema permet un codi molt més ordenat i llegible i facilita la feina alhora de buscar errors ja que cada part del codi està separada. També és molt útil quan es vol fer una nova funció o característica, ja que el que s'ha de fer és crear un nou mòdul per aquesta característica, amb el seu controlador, la seva vista i el seu model. A continuació veurem un exemple d'aquesta separació dels mòduls. Els mòduls es declaren amb la funció `angular.module()`. Agafem el mòdul de control d'accés i n'analitzem les diferents parts:

a) *login-form.html*: És la vista, conté el codi HTML amb el formulari per iniciar la sessió.

```
<div class="container" ng-if="!isLoggedIn">
  <div class="login-container">
    <div id="output"></div>
    <div class="form-box">
      <form action="" method="">
        <input name="user" type="text" placeholder="username" ng-
model="userCredentials.username">
        <input placeholder="password" type="password" ng-
model="userCredentials.password">
        <div ng-show="registerUserMode">
          <input type="email" placeholder="email" ng-model="userCredentials.email">
          <input type="text" placeholder="First Name" ng-
model="userCredentials.first_name">
          <input type="text" placeholder="Last Name" ng-
model="userCredentials.last_name">
          <button class="btn btn-info btn-block register" type="submit" ng-
click="doRegister()">Register</button>
        </div>
        <div ng-hide="registerUserMode">
```

```

        <button class="btn btn-info btn-block login" type="submit" ng-
click="doLogin()">Login</button>
    </div>

    <span class="register-link" ng-click="showRegistrationForm()">Crea un compte
de coordinador</span>
</form>
</div>
</div>
</div>

```

b) *app.js*: conté el controlador i el model, s'encarrega de mostrar el formulari d'inici de sessió quan l'usuari no està autenticat i de mostrar el contingut de l'aplicació quan l'usuari ja ha iniciat la sessió.

```

angular.module('stugrApp')
    .controller('AppCtrl', ['$rootScope', '$scope', '$location', 'Userservice',
'StugrUser', 'AUTH_EVENTS', 'USER_ROLES', function ($rootScope, $scope, $location,
Userservice, StugrUser, AUTH_EVENTS, USER_ROLES) {

    $scope.userCredentials = { 'username': '', 'password': '', 'email': '',
'first_name': '', 'last_name': '' };
    $scope.registerUserMode = false;
    $rootScope.codename = "Student Groups";

    $scope.showRegistrationForm = function() { $scope.registerUserMode = true; };

    $scope.currentUser = null;
    $scope.userRoles = USER_ROLES;
    $scope.isAuthorized = Userservice.isAuthorized;
    $scope.isLoggedIn = Userservice.isAuthenticated;
    console.log(Userservice.isAuthenticated());
    $scope.setCurrentUser = function (user) {
        $scope.currentUser = user;
    };

    $scope.doLogin = function() {
        console.log("Login Credentials: ", $scope.userCredentials);
        Userservice
            .login($scope.userCredentials)
            .then(function(response) {
                console.log('Successful login', response);
                $rootScope.$broadcast(AUTH_EVENTS.loginSuccess);
                $scope.setCurrentUser(Parse.User.current());

            }, function(reason) {
                console.log('Failed Login: ', reason);
                $rootScope.$broadcast(AUTH_EVENTS.loginFailed);
            });
    };

    // Registre d'usuaris
    $scope.doRegister = function() {
        Userservice
            .register()
            .then(function(coord) {
                // Create the user
                var user = new Parse.User();

                // set the user-related properties
                for (var property in $scope.userCredentials) {
                    user.set(property, $scope.userCredentials[property] );
                }

                // Set the coordinador linker in the 'coordinador' row of type Pointer
                user.set("coordinador", coord);

                user.signUp(null, {
                    success: function(user) {
                        console.log("User " + user.id + " successfully registered!");
                    }
                });
            });
    };
}]);

```

```

$scope.userCredentials.type = "Coordinador";
Parse.Cloud.run('sendEmailWithPassword', $scope.userCredentials, {
  success: function(httpResponse) {
    console.log("Email sent. response: " + httpResponse);
  },
  error: function(error) {
    console.log("mail error: " + error);
  }
});

},
error: function(user, error) {
  console.log('failed registration: ', error);
}
});
},
function(error) {
  console.log("Could not save the link object, error: " + error.code);
});
};

$scope.doLogout = function() {
  Userservice.logout();
  document.location.reload();
};

})();

```

c) *userservice.js*, conté la lògica del programa encarregada de comunicar-se amb el servidor Parse per autenticar els usuaris. Es comunica amb el controlador, notificant-lo quan hi ha una resposta del servidor. Aquest arxiu s'anomena Servei.

```

angular.module('stugrApp')
.service('Userservice', function Userservice($rootScope, Session, USER_ROLES, $state) {

  var authService = {}

  authService.login = function(credentials) {
    return Parse.User.login(credentials.username, credentials.password)
    .then(function(user) {
      console.log(user);
      Session.create(user.sessionToken, Parse.User.current());
      return user;
    });
  };

  authService.isAuthenticated = function () {
    return !!Session.userId;
  };

  authService.isAuthorized = function (authorizedRoles) {
    if (!angular.isArray(authorizedRoles)) {
      authorizedRoles = [authorizedRoles];
    }
    return (authService.isAuthenticated() &&
      authorizedRoles.indexOf(Session.userRole) !== -1);
  },

  authService.register = function() {

    // Create a coordinador link object with the role-specific data
    var Coordinador = Parse.Object.extend("Coordinador");
    var coordLinker = new Coordinador();

    // Save the linked object
    return coordLinker.save(null);

  };

```

```

authService.createUser = function(credentials, id) {
  // These are the arrays with the types
  var typesLinkersClasses = ["Alumne", "Professor", "Coordinador"];
  var typesRowNames = ["alumne", "professor", "coordinador"];

  // Create the linker class with the id passed
  var LinkerClass = Parse.Object.extend(typesLinkersClasses[id]);
  var linker = new LinkerClass();

  // Create the user
  var user = new Parse.User();

  linker.save(null, {
    success: function(linker) {

      // set the user-related properties
      for (var property in credentials) {
        user.set(property, credentials[property]);
      }

      // Finally we set the LinkerClass to the corresponding linking row
      user.set(typesRowNames[id], linker);
      console.log("Executing line: user.set("+typesRowNames[id]+"", "+linker+"); ");

      // Signup
      return user.signUp(null);

    },
    error: function(linker, error) {
      console.log("Could not save the link object: " + error);
    }
  });
};

authService.logout = function() {
  Parse.User.logout();
  Session.destroy();
  $state.go('home');
  $rootScope.$apply();
};

return authService;
});

```

d) Finalment, a la declaració del mòdul, tenim codi vari com l'assignació de constants o la definició de la ruta per accedir a la pàgina d'inici de sessió.

```

.constant('AUTH_EVENTS', {
  loginSuccess: 'auth-login-success',
  loginFailed: 'auth-login-failed',
  logoutSuccess: 'auth-logout-success',
  sessionTimeout: 'auth-session-timeout',
  notAuthenticated: 'auth-not-authenticated',
  notAuthorized: 'auth-not-authorized'
})
.constant('USER_ROLES', {
  coord: 'coord',
  professor: 'professor',
  alumne: 'alumne'
});
$stateProvider
.state('anon.login', {
  url: '/login',
  templateUrl: 'views/logIn.html',
  controller: 'AppCtrl',
  controllerAs: 'app'
});

```

Aquest és un exemple de la implementació d'una funció de l'aplicació. Per a qualsevol altra funció el procés seria el mateix: creació d'un mòdul i la programació de cadascuna de les parts.

## 7. Desplegament

Per a poder veure l'aplicació en marxa cal compilar els arxius i posar-la en un servidor que s'encarregui de respondre les peticions dels navegadors web dels usuaris. Compilar significa, en termes de computació, transformar els arxius escrits en llenguatge de programació a llenguatge de màquina que l'ordinador pugui entendre. Cal fer un aclariment, però: en aquest cas utilitzo el llenguatge de programació Javascript, que és un llenguatge interpretat. Això significa que l'ordinador llegeix i entén directament el programa en javascript sense necessitat de transformar-lo a llenguatge de màquina. Per aquest motiu, només en aquest cas en concret, la compilació es tracta simplement de la reducció de mida dels arxius de codi per tal que ocupin menys espai.

### 7.1. Fases de desenvolupament i servidors

El procés de compilar els arxius i penjar l'aplicació en un servidor s'anomena desplegament o *deployment* en anglès. Hi ha tres fases en què es pot trobar l'aplicació:

- a) Fase de desenvolupament, els arxius de l'aplicació es troben a l'ordinador del programador i s'utilitza un servidor local per veure l'aplicació en funcionament. Només es pot accedir a aquest servidor local des del mateix ordinador on s'està desenvolupant. Aquest funcionament permet veure els canvis que es realitzen a l'aplicació molt ràpidament, perquè els arxius no estan compilats encara.
- b) Fase d'estacionament o *staging*, en aquesta fase es posa l'aplicació en un servidor a la xarxa i es compilen arxius. Tot i això, el servidor està pensat per utilitzar-lo durant el desenvolupament, ja que aquesta fase està pensada per quan l'aplicació està en proves. Per això la modificació del codi és ràpida.
- c) Fase de producció, en aquest punt l'aplicació ja està preparada per funcionar en un entorn real i per tant es publica en un servidor amb recursos reals. Està completament desplegada.

### 7.2. Accés a l'aplicació i el codi

El següent pas per a la meua aplicació és doncs posar-la en un servidor d'estacionament. Aquest servidor d'estacionament s'anomena *Heroku*, és gratuït si s'utilitza pocs recursos (memòria, ample de banda, emmagatzematge). L'accés a l'aplicació en el servidor d'estacionament és a través de l'adreça <https://studentgroups.herokuapp.com>. He activat un compte de professor per accedir a l'aplicació:

Nom d'usuari: **abasart**

Contrasenya: **professora1234**



Durant aquesta fase es poden acabar d'incorporar característiques a l'aplicació però a la vegada es poden provar i utilitzar aquelles característiques que ja estiguin en funcionament. L'actualització del codi és fàcil i ràpida gràcies al sistema de control de versions *git*. Amb tan sols un comandament a la consola el servidor de estacionament s'actualitza amb la última versió del codi que tinc al meu ordinador.

El control de versions també em permet publicar el codi a una pàgina web de projectes de codi obert, de manera que també es pot accedir a tot el codi de l'aplicació. La pàgina on està publicat el codi es diu Github i és de les més populars per a projectes de codi obert. Es pot accedir al codi a través de l'adreça [\*\*https://github.com/hulehule20/Student-Groups-TdR\*\*](https://github.com/hulehule20/Student-Groups-TdR). El fet que l'aplicació sigui de codi obert i, per tant, permeti les contribucions de tothom, és una característica que es posa a favor que l'aplicació tingui èxit, ja que a partir del punt en què es publica el codi passa a ser una feina de tots aquells que hi participin, i no tan sols d'una sola persona o una organització. Aquesta és una característica que comparteix amb la plataforma Moodle, que també és lliure.

## 8. Conclusions

He realitzat aquest treball perquè m'agrada el desenvolupament de software i volia crear una aplicació que sigués útil. Així doncs, caldrà esperar veure l'aplicació en funcionament per treure'n les conclusions més importants. Es començarà amb una prova pilot amb pocs alumnes i algunes característiques per veure com és la rebuda per part dels usuaris. L'objectiu final és que funcioni completament als servidors de l'institut.

Valoro positivament el treball ja que m'ha permès posar en pràctica els coneixements que havia adquirit fins ara de programació, a la vegada que n'he adquirit molts de nous. El fet d'haver de transmetre de manera escrita alguns d'aquests coneixements també m'ha ajudat a convertir-los en idees més ben estructurades.

Per la part negativa, el fet d'utilitzar sistemes amb els quals no havia treballat mai i a causa de la complexitat de l'aplicació m'he trobat amb una gran quantitat de problemes durant el procés de desenvolupament que no m'han permès acabar l'aplicació del tot a temps. Tot i això seguiré amb el desenvolupament fins arribar al producte ideat.

## 9. Bibliografia

### Articles de pàgines i llocs web

HENGVELD, Gert. *Techniques for authentication in AngularJS applications*. Opinionated AngularJS, 2014 [<https://medium.com/opinionated-angularjs/techniques-for-authentication-in-angularjs-applications-7bbf0346acec>]

- *Scalable code organization in AngularJS*. Opinionated AngularJS, 2013
- *Advanced routing and resolves*. Opinionated AngularJS, 2013

ENRIQUE BOLANOS, Jose. *5 Tips for Using Parse with AngularJS*. Slidebean Blog, 2013 [<http://blog.slidebean.com/blog/2013/11/5/5-tips-for-using-parse-with-angularjs>]

*AngularJS API Docs*, mantinguda per la comunitat de desenvolupadors. [<https://docs.angularjs.org/api>]

MUTYALA, Chaitanya. *Fast prototyping — A simple authentication system using Yeoman, AngularJS and parse.com*. Medium, 2014 [<https://medium.com/@chytanya/fast-prototyping-a-simple-authentication-system-using-yeoman-angularjs-and-parse-com-49b01e99051d>]

MONGE, Eddie [et. al.] *AngularJS generator*. Github, 2015 [<https://github.com/yeoman/generator-angular>]

RAJ, Jay. *Creating a Web App From Scratch Using AngularJS and Firebase*. Envato Tuts+ Code Tutorial, 2014 [<http://code.tutsplus.com/tutorials/creating-a-web-app-from-scratch-using-angularjs-and-firebase--cms-22391>]

Vllaweb. *L'Educat 1x1, radiografia d'un projecte*. VilaWeb, 17 de Febrer de 2011 [<http://www.vilaweb.cat/noticia/3849329/20110217/leducat-1x1-radiografia-projecte.html>]

JALIM, Steve; CHOUDHARY, Neha. *What is a software framework?*. StackOverflow, 2010 [<http://stackoverflow.com/questions/2964140/what-is-a-software-framework>]

MOTTO, Todd. *AngularJS Tutorial: A Comprehensive 10,000 Word Guide*. Airpair [<https://www.airpair.com/angularjs>]

NAKSTAD, Frederik. *UI-Router in angular-client-side-auth*. A Modest Proposal, Febrer 2014  
[<http://www.frederiknakstad.com/2014/02/09/ui-router-in-angular-client-side-auth/>]

## **Annexos**

Annex 1: Glossari Terminològic	30
Annex 2: Resultats Enquesta	31
Annex 3: Disseny Aplicació	33

## Annex 1: Glossari Terminològic

- **Framework:** En termes de programació, un framework és un tipus de llibreria de software, en aquest cas una abstracció de codi comú entre diferents aplicacions que proveeix funcions específiques i reutilitzables. Per posar un exemple a la vida real, imagina que s'ha de tallar un tros de paper de 5cm per 5cm. Es podria mesurar i tallar perfectament. Però ara imagina que se n'han de tallar 1000 trossos. Segurament construiries un marc amb aquestes dimensions que t'ajudaria a anar més ràpid i a no haver de mesurar cada tros. Aquest marc, doncs, en programació, seria un framework.

- **SDK (Software Development Kit):** Un SDK és un conjunt d'eines de desenvolupament de software que permeten al desenvolupador crear aplicacions per a un sistema o una plataforma en concret, ja pot ser un tipus de consola, un sistema operatiu, un dispositiu mòbil. L'organització que crea el sistema és la que s'encarrega de fer i distribuir el SDK entre els desenvolupador interessats en fer aplicacions per aquest sistema.

- **Javascript:** És un llenguatge de programació interpretat i orientat a objectes utilitzat en pàgines web. Va néixer el 1995 a Netscape, i el 1997 va ser adoptat com un estàndard.

- **MVC:** Les sigles en anglès de Model View Controller, que en català signifiquen Model Vista Controlador.

- **HTML:** Les sigles en anglès de HyperText Markup Language. És un llenguatge d'etiquetes que s'utilitza per al disseny de pàgines web.

- **PHP:** És un llenguatge de programació basat en etiquetes utilitzat per a desenvolupament de servidors web. Actualment és el més utilitzat, el podem trobar a les pàgines aplicacions web més populars d'Internet (com per exemple Facebook).

- **iOS:** És el sistema operatiu creat per Apple que utilitzen els dispositius mòbils d'aquesta companyia. Està basat en l'arquitectura Unix.

- **Objective-C:** Llenguatge de programació orientat a objectes basat en C que s'utilitza per fer aplicacions per a iOS i ordinadors Mac.

- **Swift:** Llenguatge de programació potent, ràpid i intuïtiu creat per Apple l'any 2014. S'utilitza per a fer aplicacions per a les plataformes d'aquesta companyia. Des del novembre de 2015 Swift és un llenguatge de codi obert i tots els desenvolupadors poden col·laborar en la seva evolució.

## **Annex 2: Resultats Enquesta**

Primera Part: Valoració sistema actual

Segona Part: Idees per la nova plataforma







*“Thank you for reading this far. We apologize for the complexity. Modeling relationships in data is a hard subject, in general. But look on the bright side: it's still easier than relationships with people.” —*

Documentació de [parse.com](https://parse.com)

*“Gràcies per llegir fins aquí. Demanem disculpes per la complexitat. Modelar les relacions entre dades és una matèria complicada, en general. Però mira el costat positiu: segueix sent més fàcil que les relacions amb les persones.”*