



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

Aplicació multiplataforma per a la gestió d'una escola musical.

MEMÒRIA PRESENTADA PER:

Juan Camús Sanz

DIRIGIT PER:

Manolo Llorca

GRAU D'ENGINYERIA INFORMÀTICA

Convocatòria de defensa: Febrer-Març 2021



Resum:

L'aplicació multiplataforma és el resultat de 6 mesos d'investigació. L'objectiu d'aquest projecte és el desenvolupament i documentació d'una aplicació web per a poder gestionar l'escola de música Tina Ferre.

Es tracta d'una intranet a xicoteta escala que tindrà que garantir l'accés tant d'estudiants com de professors del centre. Els professors, podran gestionar les seues assignatures, afegint material complementari, també podran donar avisos, pujar notes i donar alguna informació més per al estudiant. D'altra banda l'estudiant podrà visualitzar tot el material però sense poder canviar res.

Aquesta aplicació web està preparada per a qualsevol dispositiu, és a dir, té un disseny "responsive" o adaptable per a qualsevol tamany de pantalla, d'aquesta forma garantim que els usuaris que l'utilitzen puguen accedir des de diferents tipus de dispositius com ordinadors fins telèfons mòbils, o tabletas.

Es tracta d'un projecte de desenvolupament software des de zero, i s'han utilitzat principalment els Frameworks, Angular per al Front-End, i el Framework Laravel per al Back-End, per aconseguir l'arquitectura anomenada RESTFul API.

Aquesta escola, no tenia cap plataforma educativa, per la qual cosa en requerien d'una per poder fer front a aquest desenvolupament amb línia.

Principals requeriments d'aquesta aplicació són:

- Framework de desenvolupament Back-End Laravel.
- Framework de desenvolupament Front-End Angular.
- Aplicació web Responsive, que s'adapte a qualsevol dispositiu.

Paraules Clau:

Aplicació WEB, escola, Intranet, PHP, RESTFul API.

Summary:

The multi-platform application is the result of 6 months of research. The main objective of this project is to develop and document a web application to manage the Tina Ferre music school.

This is a small-scale intranet that will have to guarantee access for both students and teachers at the centre. The teachers will be able to manage their subjects, adding complementary material, they will also be able to give warnings, to raise notes and to give some additional information for the student. On the other hand, the student will be able to view all the material without modifying the content.

This web application is ready for any device, it has a responsive or adaptable design for any screen size, this way we guarantee that users who use it can have access to different types of devices, as they can access from computers to mobile phones, or tablets.

This is a software development for scratch project, it is developed by Angular Frameworks, in the Front-End, and Laravel Framework for the Back-End have been used primarily to achieve the architecture called the RESTful API.

This school did not have any educational platform, so they needed a new platform to deal with this development online.

The main requirements of this application are:

- Laravel Back-End Development Framework.
- Front-End Angular development framework.
- Responsive web application, which adapts to any device.

Keywords:

Application Web, School, Intranet , PHP, RESTFul API.



Índex de Continguts

1.	INTRODUCCIÓ.....	6
1.1.	Presentació del projecte	6
2.	DESCRIPCIÓ	7
3.	Objectius	8
3.1.	Principals	8
3.2.	Secundaris	8
4.	ESTAT DE L'ART	9
4.1.	Estat de l'Art tecnològic	9
4.1.1.	Google Classrom.....	10
4.1.2.	Edmodo	10
4.1.3.	Moodle	11
4.2.	Estat de l'Art legislatiu.....	12
4.2.1.	Protecció de dades en pàgines web.....	12
5.	ANTEPROJECTE.....	14
6.	CONTINGUTS	15
7.	METODOLOGIA.....	17
8.	ARQUITECTURA DE L'APLICACIÓ	18
9.	PLATAFORMA DE DESENVOLUPAMENT	20
9.1.	Programari.....	20
9.2.	Maquinari	22
10.	PLANIFICACIÓ	23
11.	ANÀLISI I DISSENY	24
11.1.	Estructura de l'API REST	24
11.1.1.	Capa de Seguretat o Middleware.....	25
11.1.2.	Capa de Lògica de Dades.....	27
11.1.3.	Capa de la base de dades	27
11.1.4.	Estructura de carpetes Back-End	30
11.2.	Disseny de l'aplicació web.....	31
11.2.1.	Estructura de Carpetes Frontend	31
11.2.2.	Disseny de Mockups versió escriptori	33
11.2.3.	Disseny de mockups versió tablet	45
11.2.4.	Disseny de mockups versió smarthpone.....	46

11.2.5.	Diagrama de funcionament.....	47
12.	IMPLEMENTACIÓ.....	49
12.1.	Implementació del Back-End.....	49
12.1.1.	Instal·lació Back-End.....	49
12.1.2.	Rutes i comunicacions internes.....	52
12.1.3.	Middleware	53
12.1.4.	Models.....	54
12.1.5.	Controladors.....	56
12.1.6.	Migracions.....	58
12.1.7.	Seeders.....	60
12.2.	Implementació del Front-End.....	61
12.2.1.	Instal·lació del Front-End.....	61
12.2.2.	Servicis.....	62
12.2.3.	Token Interceptor.....	64
12.2.4.	Route Guards.....	65
12.2.5.	Routing Module.....	66
12.2.6.	Internacionalització	66
12.2.7.	Seguretat	67
12.3.	Codi Font	69
13.	Pressupost.....	70
13.1.	Maquinari o costos materials.....	70
13.2.	Programari o costos tècnics	71
13.3.	Cost Personal.....	71
13.4.	Pressupost Final	72
14.	CONCLUSIONS.....	73
14.1.	Conclusions.....	73
14.2.	Treball futur.....	73
15.	BIBLIOGRAFIA.....	74
16.	Índex d'Il·lustracions	75
17.	ANNEXOS.....	77
17.1.	Annex token service	77
17.2.	Annex instal·lació Xdebug	78



1.INTRODUCCIÓ

1.1. Presentació del projecte

El projecte d'aquesta memòria es centra en l'Escola Tina Ferre de l'Agrupació Musical La Nova situada a la localitat de Banyeres de Mariola. Es tracta d'un projecte web, que busca solucionar una carència de serveis als membres de la comunitat educativa (professors, alumnes, administratius i famílies) entre altres.

Aquesta web facilitarà informació, formació, recursos didàctics, ferramentes per a la comunicació interpersonal, notes, faltes, horaris, matrícula..., és a dir, un portal web per a la comunitat educativa.

Avui en dia, l'escola de música no ofereix cap tipus de recurs en línia ni disposa de ferramentes per a fer-ho. Es va plantejar la creació d'un portal educatiu on indistintament del dispositiu que s'use es puga accedir i d'aquesta forma aquests servicis que ofereixen el portal estarien sempre accessibles des de qualsevol moment i lloc.

A partir d'aquest plantejament, es va començar la tasca de definir les necessitats, funcionalitats i requisits indispensables d'aquest portal. Es van establir diferents reunions amb el professorat tècnic de l'escola per garantir els seus requeriments i establir la seua funcionalitat.

D'aquestes reunions es van plantejar les possibles solucions proposades en aquesta memòria.

També cal esmentar, que avui en dia existeixen plataformes dedicades exclusivament a portals educatius, en l'apartat de l'estat de l'art es discutirà per que no s'ha seleccionat cap i s'ha decidit fer un portal des de zero.

En les següents pàgines es presentaran els objectius, metodologies i ferramentes utilitzades per al desenvolupament d'aquesta aplicació.

2. DESCRIPCIÓ

Es proposa com a Treball Final de Grau, en avant TFG, el desenvolupament d'una aplicació web multiplataforma per a la gestió d'una escola musical. Aquesta aplicació web tindrà tres rols. Rol Estudiant, Rol Professor i Rol Administratiu. Es crea aquesta aplicació per a ajudar al desenvolupament intel·lectual dels alumnes i ajudar als professors a organitzar millor les seves classes i metodologies.

El projecte té com a requisit ser totalment escalable, és a dir, es podran afegir diferents funcionalitats al llarg de futures necessitats. Per a la primera versió ens centrarem en les funcionalitats principals que es mostren a continuació, ordenades de major a menor importància:

- Crear nous usuaris i assignar el Rol (Estudiant, Professor o Administratiu).
- Donar d'alta noves assignatures, per a qualsevol curs.
- Assignar assignatures amb professors i alumnes.
- Crear alertes de professors per a una assignatura en concret.
- Pujar recursos des del Rol professor i poder visualitzar-los en el Rol alumne.
- Habilitar un espai on l'alumne pugui comunicar-se amb el professor.
- Donar informació sobre l'assignatura que s'està impartint.

En un món tant globalitzat, l'ús d'Internet, ens ha ensenyat que hem de tindre amb compte el que allotgem a Internet, per tant, s'ha escollit utilitzar el Framework Laravel per securitzar l'aplicació, i fer-la robusta per tal de que aquesta aplicació siga fiable i segura.

Com bé diu el títol d'aquest TFG, aquest projecte s'ha centrat en l'orientació d'utilitzar diferents dispositius per poder accedir a l'aplicació, s'ha escollit el Framework Angular per crear una part web intuïtiva i que no requereisca de cap nivell alt de coneixements per poder utilitzar-se.

Aquest projecte, també aportarà el codi de l'aplicació, i una breu guia de com hi ha que implementar aquest codi per a que funcione correctament.

Per concloure, es pretén realitzar el disseny i desenvolupament d'una web per a l'escola de manera didàctica.

El nom escollit per a la plataforma és: "Escola Tina Ferre", ja que és el nom que rep l'escola actual.

3. Objectius

3.1. Principals

Els objectius a aconseguir principalment són:

- Crear una interfície web per a l'administració per tal de gestionar tant alumnes com professors i assignatures.
- Documentar cada fase del desenvolupament d'aquesta aplicació, des de l'idea del projecte passant per les distintes fases de disseny, desenvolupament i publicació d'aquesta.

L'objectiu principal a aconseguir és crear el portal on els alumnes i professors puguen disposar d'ell les 24 hores. Per poder garantir aquest projecte s'ha de garantir els següents objectius:

- Crear primerament un Back-End robust, és a dir, perquè el portal web funcione de manera òptima les 24 h del dia, els 365 dies a l'any s'ha de garantir que el Back-End no falle.
- Crear una aplicació, escalable, amb el pas del temps.

3.2. Secundaris

Objectius addicionals, per enriquir l'aplicació.

- Crear una interfície web on els alumnes puguen accedir a tota la documentació relacionada amb les seues assignatures matriculades.
- Crear una interfície web on els professors realitzen gestions educatives com: pujar notes, recursos, o fer algun tipus de notificació als pares.
- Aquestes interfícies web ha de respectar el disseny Responsive en tot moment.
- Aprendre més sobre el coneixement de les tecnologies WEB de desenvolupament d'aquest projecte Laravel en el Back-End i Angular en el Front-End.
- Estructurar el desenvolupament d'aquesta web en temps emprat.
- Pressupost orientatiu del qual costaria aquesta web.

4. ESTAT DE L'ART

A continuació es presenta l'estat de l'art que alhora està subdividit en tres parts. Estudiarem les diferents aplicacions educatives que existeixen al mercat avui en dia en l'estat de l'art tecnològic, estudiarem també els requisits de privacitat que tota aplicació publicada en la Unió Europea ha de tindre en compte en l'Estat de l'Art legislatiu i per últim explicarem breument per què s'ha decidit prendre el camí d'aquest projecte i crear una aplicació des de zero.

4.1. Estat de l'Art tecnològic

En primer punt, s'analitzaran diferents plataformes educatives virtuals actuals que s'utilitzen en altres escoles o conservatoris per gestionar alumnes de l'àmbit musical. S'entén com plataforma educativa virtual un programa software que engloba diferents tipus de ferramentes destinades a fins docents. Són programes que permeten fer tasques com: organitzar contingut i activitats dintre d'un curs, gestionar les matrícules dels estudiants, tindre un seguiment de treball durant el curs, resoldre dubtes, crear espais de comunicació interactiva, avaluar el progrés de l'alumne, entre altres...

Perquè una plataforma educativa, pugui ser utilitzada, deu constar de les següents ferramentes:

- LMS(Learning Management System): És un lloc on s'encontren i contacten tots els usuaris de la plataforma: alumnes, professors i personal administratiu. Ací és on són presentats els cursos als usuaris i on es realitza el seguiment del progrés de l'alumne durant el temps que dura la seua formació.
- LCMS (Learning Content Management System): És una ferramenta que ens permet la gestió i publicació dels continguts utilitzats en el curs.
- Ferramentes de comunicació: Afavoreixen la participació dels estudiants creant espais dedicats al treball en comú i intercanvi d'informació. Normalment, es fa a través de xats, correus electrònics...
- Ferramenta d'administració: Ens permet la gestió de les inscripcions, donar diferents permisos d'accés dintre de la plataforma als diferents usuaris.

Existeixen diferents tipus de plataformes educatives, entre elles s'encontren les plataformes educatives comercials, que han sigut creades per empreses amb fins lucratiu, entre les quals destaquen WebCT o FirstClass i altres plataformes que s'han creat amb programari lliure sense cap tipus de fins lucratiu com Classroom, Edmodo i Moodle.

Avui en dia, amb tot el que està succeint, amb la pandèmia global, i tot el que ha ocorregut amb el virus SARS-CoV-2 o Covid-19 ens ha demostrat que necessitem uns



mitjos d'aprenentatge complementari ja que l'educació presencial a vegades no és possible i necessitem un recurs alternatiu per poder seguir aprenent.

En aquest TFG ens centrarem en les plataformes de codi font lliure.

4.1.1. Google Classroom

La plataforma Google Classroom, és una ferramenta creada per Google en 2014, i destinada exclusivament al món educatiu. La missió d'aquesta plataforma es basa en permetre gestionar una aula de forma col·laborativa a través d'Internet.

Totes les opcions d'aquesta ferramenta estan associades a un compte de Google, és a dir, s'ha de disposar d'un compte de Google per poder accedir-hi.

Aquesta plataforma és un servici gratuït, i entre les possibles avantatges que ofereix es troben la de crear documents, compartir informació en diferents formats, guardar reunions i realitzar-les virtualment.

Dintre dels principals objectius d'aquesta plataforma destaquem el de simplificar i ajudar en la distribució de material educatiu, ja siga en format de vídeo, àudio, PDF o avaluacions. Aquesta plataforma, forma part d'una iniciativa més gran coneguda com a Google for Education.

També cal destacar els seus desavantatges ja que aquesta plataforma, no és considerada una plataforma LMS, sinó que més bé, seria considerada com Blended Learning, una ferramenta per a ajudar a l'educació de manera presencial.

Per últim cal destacar que aquesta plataforma ha sigut acusada per emmagatzemar, per a fins publicitaris, historials de recerques i dades d'usuaris de "Classrom", violant així, la privacitat implícita de la plataforma. Aquesta acusació ha sigut fortament criticada per l'opinió pública.

Aquesta plataforma es distribueix baix la llicència "Creative Commons tipus BY-NC-SA", és a dir, el qui utilitze aquesta llicència té dret a compartir i adaptar sempre i quan no siga per a fins comercials i si modifica el seu codi font deu de compartir-ho baix la mateixa llicència original.

Aquesta plataforma actualment s'utilitza en algunes assignatures del Conservatori Municipal de Banyeres de Mariola.

4.1.2. Edmodo

La plataforma Edmodo és una plataforma educativa orientada a la part acadèmica del e-learning pel que està fortament enfocada a les institucions educatives com Escoles i Universitats.

Aquesta aplicació està definida com a una plataforma K-12, pel que el seu ús està més orientat a escoles de nivell primària (fins 12 anys).

Molts usuaris consideren a Edmodo com una xarxa social educativa, quasi com una extensió de facebook per a l'escola.

Edmodo no és considerada una plataforma LMS, ja que no disposa de tots els estàndards mínims requerits. Aquesta plataforma està dissenyada per a la comunicació entre alumnes, professors i familiars dels alumnes.

Entre les característiques d'aquesta web destaquen: disponibilitat de descarregar per a Android e IOs, vista prèvia dels fitxers, integració en aplicacions de tercers com Google Drive, possibilitat de creació de subgrups i grups d'estudiants i professors, calendari, enquestes. Aquesta plataforma està basada en 3 rols: Alumne, Pares i Professors. Per últim disposa d'una característica de gamificació d'insígnies i premis que els professors poden atorgar als alumnes d'acord amb el seu esforç.

Entre els desavantatges que ofereix aquesta aplicació és que no és una plataforma LMS, i a més existeixen dues versions, una gratuïta limitant les característiques descrites anteriorment i una altra més completa on estan totes les característiques descrites per un cost a l'any.

Aquesta plataforma es distribueix baix la llicència EULA abreviatura de "End-User License Agreement" .

Actualment s'usa al Conservatori Superior de València "Joaquín Rodrigo".

4.1.3.Moodle

La plataforma Moodle és una plataforma educativa d'accés lliure i gratuïta.

Entre les característiques més importants de Moodle destaquen: és una ferramenta estable i de confiança, intuïtiva i fàcil d'usar, sempre actualitzada, flexible i personalitzada, escalable a qualsevol tamany, ubiqua i accessible des de qualsevol dispositiu, robusta, segura i privada i consta de funcionalitats ampliables mitjançant connectors i complements.

La plataforma s'estructura al voltant de tres rols: Professors, Alumnes i Administradors.

Aquesta plataforma és capaç de gestionar usuaris, accessos i rols, dissenyar una estructura pedagògica, dissenyar recursos didàctics, crear espais d'aprenentatge comú, establir vies de comunicació entre el professor i alumnes, avaluar els alumnes i generar informes, entre altres funcions.

Alguns dels seues desavantatges són que el seguiment de cada estudiant es dificulta, és a dir, falta millorar la interfície de manera més senzilla, també cal destacar que segons on instal·les Moodle pot haver problemes de seguretat associats.



Es distribueix baix la llicència GPL (General Public License), el que significa que qualsevol institució pot fer ús d'ell i modificar-lo a les necessitats, sense haver de pagar per ell.

Aquesta plataforma actualment s'usa al Conservatori de Villena.

4.2. Estat de l'Art legislatiu

En aquest apartat ens centrarem, en la llei de protecció de dades que existeixen en l'actualitat per als serveis web per a tota la Unió Europea.

4.2.1. Protecció de dades en pàgines web

Segons les estadístiques realitzades per l'Agència Espanyola de Protecció de Dades, els internautes mostren una alta preocupació per el tractament de la informació personal.

Segons l'institut nacional d'estadística en Espanya les xifres d'usuaris intensius d'internet està al voltant de 24 milions de persones, que són quasi el 69% de la població.

Els usuaris coincideixen en:

- Més del 82% d'usuaris prioritza el tema del tractament de dades personals, un tema de gran importància.
- Tindre dret a la intimitat i privacitat en internet.
- El nombre de denúncies de la llei de protecció de dades per falta d'informació en la política de "cookies" ha augmentat en els últims anys.

En desembre de 2018 va entrar en vigor una nova Llei de Protecció de Dades substituint la que ja hi havia. L'objectiu d'aquesta llei LOPDGDD o Llei Orgànica de Protecció de Dades i Garantia de Drets Digitals, és adaptar la legislació espanyola a la normativa europea, definida pel Reglament General de Protecció de Dades (RGPD), vigent des del 25 de maig de 2018.

Aquesta nova llei té com a principals punts, els següents:

- **Àmbit:** Al tractar-se d'una normativa europea l'àmbit compren a tots els països membres de la Unió Europea.
- **Aplicació:** Afecta a totes les empreses i institucions dels països de la Unió Europea, i les que realitzen operacions comercials dintre d'Europa, encara que estiguen establides fora de la Unió Europa.

- **Avís de privacitat:** Aquesta llei exigeix també a les empreses, a una revisió de les seues polítiques de privacitat. Hauran d'explicar la base legal per al tractament de dades, el període de temps de retenció de les dades i que els interessats puguin dirigir les seues reclamacions a les autoritats de protecció de dades.
- **Consentiment explícit:** Per a utilitzar les dades de caràcter personal cal comptar amb l'autorització expressa de l'usuari. Les empreses deuran demostrar que l'usuari ha donat el consentiment per a l'ús de les seues dades. Així que el consentiment deu ser explícit, específic i verificable.
- **Dret l'oblit i rectificació:** La llei dona més suport a l'usuari per facilitar més elements de control i mitjos per a exercir el dret.
- **Informació expressa:** La llei obliga a les empreses a mostrar de forma clara i accessible totes les seues dades legals, així com les seues polítiques de privacitat i l'ús de les dades personals que realitzen a través d'Internet.
- **Responsabilitat activa:** Les empreses han d'adoptar mesures per a assegurar el compliment de la nova llei de protecció de dades, això significa, disseny de la pàgina web, avaluacions d'impacte, informes sobre protecció de dades, revisions de polítiques...

5. ANTEPROJECTE

Després d'analitzar les plataformes més conegudes que existeixen avui en dia al mercat educatiu, i saber les lleis que les regeixen, vaig decidir en un primer moment buscar més informació sobre la plataforma educativa Moodle. El que més em va cridar l'atenció d'aquesta va ser que al tractar-se d'una llicència de distribució GPL, podia modificar-la perquè s'adaptara al model que actualment estem impartint a l'escola Tina Ferre.

Els alumnes potencials d'aquesta escola generalment són alumnes de 8 anys fins adults. És a dir, existeixen dos grups: gent major per vocació que vol aprendre i xiquets l'edat mínima dels qual per apuntar-se són 8 anys i que tenen curiositat per la música.

Com ja he dit, en un primer moment vaig pensar que la solució més encertada era la d'adaptar Moodle als requeriments de l'escola. Al instal·lar-lo, i reconèixer la interfície, i tots els fitxers que formaven l'ecosistema de Moodle em vaig adonar que aquest model no era per a l'escola, ja que disposava de moltes funcionalitats que no anaven a utilitzar-se realment i pel que fa al disseny que presentava podia ser confús per a l'usuari final, ja que l'experiència de la interfície podria donar pas a confusions.

Després de provar-ho, és quan em vaig reunir en l'equip docent i els vaig explicar de fer un projecte des de zero. Fer tant el Back-End com Front-End per a adaptar-lo a les necessitats que requeria l'escola. Així aprofitant que ja tenim un servidor on mostra la pàgina web de l'agrupació, es podria afegir un enllaç a l'escola i donar-li més funcionalitat a la web.

Per poder desenvolupar l'aplicació educativa, tenint en compte els requeriments del client, en aquest cas l'equip docent de l'escola, es van plantejar diferents arquitectures, fins que es va decidir dissenyar la plataforma on l'estil de l'arquitectura es basara en el protocol API que usa crides http per a accedir a l'ús de les dades. Aquesta tecnologia REST (Representational State Transfer) al ser crides http resultava ser molt flexible i escalable, per a que en un futur, no haver de refer-ho tot, garantint així la reutilització de codi.

6. CONTINGUTS

L'aplicació web constarà d'una composició clàssica respecte a l'estructura web, és a dir, en terminologia web es coneix com: un header, un sidenav, un body, i un footer.

- **Header:** Aquest component contindrà la barra de navegació on es mostraran les assignatures, la imatge de l'escola, un botó de perfil i un botó de logout.
- **Sidenav:** Aquest component, s'activarà quan es seleccione una assignatura i podrem navegar a les seccions: inici, recursos, notes, contacte, i sobre.
- **Body:** En aquest component, és on es mostrarà totes les funcionalitats de l'aplicació, és a dir, en aquest apartat, es visualitzaran totes les dades, formularis, i interaccions del sistema.
- **Footer:** Aquest component tractarà de recollir la informació legal i apareixeran diferents links, als diferents compters de les xarxes socials de l'escola.

L'estructura descrita anteriorment es respectarà durant tot el projecte en les diferents pàgines. Anem ara, a esmentar els continguts de les pàgines principals.

- **Inici:** Indiferentment del rol en què t'identifiques en l'aplicació tindràs un missatge de benvinguda i un missatge que l'Administrador de la web podrà fer, generalment estarà buit, sols per a esdeveniments o notifikacions importants a tota la comunitat educativa de la plataforma.
- **Assignatura:** Si seleccionem alguna assignatura que està continguda en el header, podrem anar fins a la informació d'aquesta i automàticament ens mostrarà informació que el professor vulga divulgar a l'estudiant.
- **Perfil:** Si seleccionem sobre l'usuari et dirigirà a l'apartat de perfil d'usuari on podràs veure totes les dades personals que utilitza l'aplicació.
- **Recurs:** Si seleccionem una assignatura i seleccionem l'apartat de recurs, els estudiants podran visualitzar i descarregar material didàctic que el professor haja pujat per a aquesta assignatura. Si pel contrari eres professor podràs afegir nous materials, així com eliminar-ne, també canviar l'estat de privacitat de públic i privat per tal de que els alumnes associats a l'assignatura puguin veure-ho o no.
- **Contacte:** Si hem seleccionat una assignatura, i seleccionem contacte, els estudiants podran contactar amb el professor afegint el tema i el missatge. El professor podrà llegir-ho i respondre't.
- **Sobre:** Si hem seleccionat una assignatura i polsem en sobre, els professors podran afegir les puntuacions, com avaluaràs, i diferents aspectes relacionats amb l'assignatura, podràs canviar l'estat de la visibilitat per tal de fer-ho públic o

privat als alumnes. Si eres estudiant, podràs visualitzar sols el que el professor elegisca.

- **Notes:** En aquest apartat, ens mostrarà les notes actuals dels diferents exàmens que l'alumne haja realitzat. Si eres professor podràs afegir noves notes als teus alumnes.
- **Administrador-Usuaris:** Si t'autentifiques amb el rol Administrador, accediràs a aquesta pantalla on podràs eliminar, o afegir nous usuaris de diferent rol (Professor, Estudiant o Administrador).
- **Administrador-Assignatura:** Si t'autentifiques amb el rol Administrador, podràs gestionar les assignatures tant per cursos com totes, i podràs modificar-les, eliminar-les o crear de noves.
- **Administrador-Estudiant-Assignatura:** Si t'autentifiques com a Administrador, aquesta pantalla et permet relacionar estudiants en assignatures, es a dir, el gestionament bàsic de les assignatures i alumnes.
- **Administrador-Professor-Assignatura:** Si t'autentifiques com a Administrador, aquesta pantalla et permet relacionar els usuaris de tipus Professor amb les assignatures que impartisca, serveix per a gestionar i pots afegir o eliminar relacions.

7. METODOLOGIA

A partir dels diferents requeriments inicials que hem plantejat, perquè l'aplicació tinga la funcionalitat desitjada s'ha iniciat el començament de la documentació i la planificació d'aquest projecte, aquesta serà la primera fase de la qual iniciarem el projecte.

En la informació obtinguda detalladament amb diferents reunions amb el claustre de l'escola al llarg del mes de Juny de 2019, posem inici a la segona fase, on s'efectuarà un anàlisi i disseny dels requeriments, es definiran l'arquitectura adoptada per l'aplicació, les tecnologies usades, les pantalles del client i el flux de navegació de l'aplicació.

Referent a les pantalles del client per a realitzar els prototips de pantalles, cal destacar que han sigut basats en la llei de la bona forma o Principi de Simplicitat. Aquesta llei ens indica:

“La Ley de la buena forma remite a un principio de organización de los elementos que componen una experiencia perceptiva y que los gestaltistas llamaron Pregnancia (Prägnanz).

Este mecanismo permite reducir posibles ambigüedades o efectos distorsionadores, buscando siempre la forma más simple o la más consistente; en definitiva, nos permite ver los elementos como unidades significativas y coherentes. Es llamado también el Principio de Simplicidad.”

Una vegada definit com anem a procedir, cal destacar que es treballarà de forma local, amb ajuda d'alguna eina per poder treballar en localhost. A més de tot açò, s'utilitzarà Git per pujar el treball al núvol i poder mantindre un control de versions, que sens dubte ens facilitarà les coses per a dur un control i per a fer front a possibles desastres que puguen succeir.

Una vegada definida la preparació de treball, és moment de desenvolupar l'aplicació web per a gestionar l'escola de música, incloent tota la documentació dels processos executats perquè aquesta aplicació funcione.

D'altra banda, en l'últim punt del projecte es faran correccions i futures millores per tal de garantir la millora i l'escalabilitat d'aquesta aplicació. D'aquesta manera deixarem una versió punt 1.0 operativa.

8. ARQUITECTURA DE L'APLICACIÓ

Aquesta aplicació està basada en l'arquitectura que es coneix com a API REST ful. Es tracta de programació d'aplicacions que s'ajusten als límits de l'arquitectura REST. Sense aquest paradigma resultaria quasi impossible tot el creixement horitzontal. Entenem per creixement horitzontal un concepte d'escalabilitat. Al ser un requisit indispensable el poder fer una aplicació escalable s'ha triat aquesta arquitectura perquè s'ajusta als requeriments.

Per a entendre un poc millor l'arquitectura REST, es defineix, com qualsevol interfase entre sistemes (Client – Servidor) que use crides HTTP (Hypertext Transfer Protocol) al Servidor per obtenir dades o generar operacions, sobre aquestes dades, poden ser en format XML o JSON, aquest últim, és el que s'ha escollit per a utilitzar en tota l'aplicació web.

Ja hem definit que la nostra aplicació web es basarà en l'arquitectura REST ful. Aquesta arquitectura necessàriament ha de ser una part Client i per altra Servidor. El Client farà les peticions al Servidor i el Servidor li respondrà en funció de la crida.

Per una part el Client s'ha escollit desenvolupar-lo en el Framework Angular, ja que ens ofereix aplicacions fàcils de mantindre, i aquest és un requisit. Per a la part del Back-End s'ha escollit el Framework Laravel perquè ens ofereix molta versatilitat.

- Angular serà l'encarregat de fer peticions http a Laravel i mostrar la informació obtinguda en la pàgina web perquè l'usuari interactue amb el navegador.
- Laravel serà l'encarregat de gestionar les peticions del client i donar resposta, a les crides http, en format JSON.

Totes les crides que es realitzen al Back-End tindran una resposta JSON, així ho indica l'estàndard.

Algunes de les característiques REST necessàries per al projecte són:

- **Protocol client/servidor sense estat:** on es tractarà cada petició HTTP de forma independent i contenint tota la informació necessària per a executar-la, el que garanteix que ni el client ni el servidor necessite recordar cap estat previ per a fer-ho.
- **Les operacions** presents en aquest projecte per a les dades del sistema REST, seran: **POST** (crear), **GET** (llegir i consultar), **PUT** (editar) i **DELETE** (eliminar).

- **Els objectes en REST sempre es manipulen a partir de la URI**, es a dir, és la URI l'identificador únic de cada recurs del sistema REST. Ens facilitarà accedir a la informació per a la seua modificació o eliminació, etc...
- **Interfase Uniforme:** és necessària per a la transferència de dades en el sistema REST, ja que s'apliquen aquestes accions concretes (POST, CREATE, PUT i DELETE) sobre els recursos, sempre i quan estiguen identificats en una URI. Aquest sistema ens brinda l'existència d'una interfase uniforme que sistematitza el procés de la informació.
- **Sistema de capes:** arquitectura jeràrquica entre els components. Cada una d'aquestes capes està sota una funcionalitat dins del sistema REST.

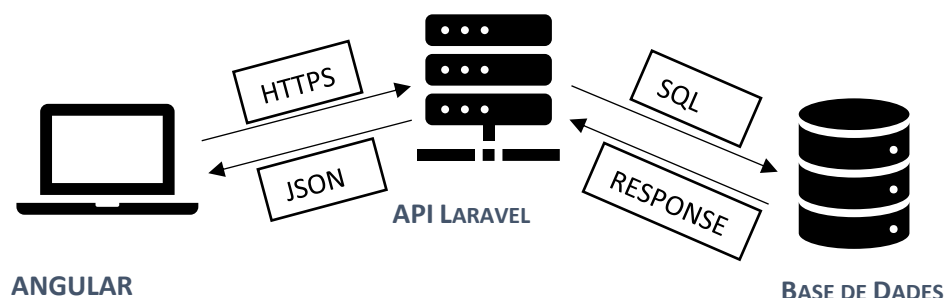
Les principals avantatges que ofereix REST per al projecte són:

1. Separació entre el client i el servidor.
2. Visibilitat, fiabilitat i escalabilitat.
3. L'API REST sempre és independent del tipus de plataforma o de llenguatge.

Els codis que retorna una petició de dades al client poden ser:

- 2XX: Petició Correcta.
- 3XX: Redireccionament.
- 4XX: Error del Client.
- 5XX: Error del Servidor.

A continuació és mostra una crida http des del client. Laravel retorna un fitxer JSON.



IL·LUSTRACIÓ 1 - ARQUITECTURA

9. PLATAFORMA DE DESENVOLUPAMENT

En aquest apartat es detallaran totes les ferramentes utilitzades per poder desenvolupar aquest projecte tant en l'àmbit de programari com de maquinari.

També per cada ferramenta de programari hi ha una breu explicació de per què s'ha utilitzat.

9.1. Programari

- **Visual Studio Code:** “És un entorn de desenvolupament integrat que proporciona facilitats al desenvolupador software per fer programes”. Aquest entorn ens ha proporcionat ferramentes potents com puga ser el sistema de depuració de codi. S'ha utilitzat per a programar tant el Front-End com el Back-End.
- **XAMP:** “és un paquet de software lliure, que consisteix principalment en el sistema de gestió de base de dades MySQL, el servidor web Apache i els intèrprets de llenguatge script PHP”. Aquest programa ha sigut clau per a desenvolupar en localhost.
- **NODE.JS:** “Node.JS és un entorn d'execució de JavaScript orientat a esdeveniments asíncrons”. Aquesta ferramenta ha sigut dissenyada per fer aplicacions en xarxes escalables. Node.JS és necessari per a poder instal·lar Angular CLI.
- **ANGULAR CLI:** Angular CLI és una ferramenta per línia de comandaments que facilita la creació, generació, execució, proves i desplegables de l'aplicació. Aquest software s'ha utilitzat activament en la programació del Front-End.
- **ANGULAR:** Es tracta del Framework de JavaScript de codi obert creat per Google per a facilitar la creació i programació d'aplicacions web d'una sola pàgina, les webs SPA. Aquest Framework s'utilitzarà per a crear el Front-End de l'escola Tina Ferre.
- **HTML:** “Hyper Text Markup Language” o més conegut com a HTML, és un llenguatge de programació que s'utilitza per a desenvolupar pàgines web. És l'encarregat de generar la part gràfica de la web. Aquest llenguatge s'utilitzarà per a crear l'apartat gràfic del Front-End de l'escola.

- **SCSS:** “Syntactically Awesome Style Sheets” o SCSS és un llenguatge d’estils, ens permet generar, de manera automàtica, fulles d’estil. S’ha utilitzat per modificar l’aspecte i format del Front-End de l’aplicació de l’escola.
- **TYPESCRIPT:** És el llenguatge utilitzat en angular. Es tracta d’un llenguatge de codi obert orientat a la programació d’objectes. Els navegadors són capaços d’interpretar TypeScript. Es tracta d’un llenguatge molt paregut a JavaScript i s’ha utilitzat durant tot el Front-End de l’aplicació.
- **LARAVEL:** Es tracta d’un Framework de codi obert basat en PHP. És simple, molt potent i té una interfície elegant. Molt utilitzat en arquitectures RESTful. S’ha utilitzat per fer el Back-End, en concret l’API de l’escola.
- **POSTMAN:** Postman és una ferramenta que és utilitzada, sobretot, per a proves d’API RESTs. Gràcies a aquesta ferramenta es poden fer proves, consumir i depurar API RESTs. Aquesta aplicació s’ha utilitzat per fer les proves de l’API de l’escola.
- **MySQLWorkbench:** Juntament amb el sistema de gestió de base de dades de Xamp, MySqlWorkBench s’ha utilitzat per exportar el model de la base de dades creat per el Framework Laravel.
- **Balsamiq:** Aquesta pàgina web s’ha utilitzat per crear esbossos o wireframes per al disseny de l’aplicació de l’escola.
- **Google Drive:** Aquesta aplicació s’ha utilitzat per guardar la documentació de la memòria a mode d’evitar pèrdues.
- **Microsoft Word:** Es tracta d’un programa orientat al processament de textos creat per Microsoft i s’ha utilitzat per redactar la memòria d’aquest projecte.
- **Gantt Project:** Aplicació d’escriptori encarregada de generar diagrames de Gantt. S’ha utilitzat per fer el diagrama de Gantt de l’aplicació.
- **Git:** És un software de control de versions, ha sigut dissenyat pensant en l’eficiència, la confiabilitat i la compatibilitat del manteniment de versions. S’ha utilitzat per pujar el codi al repositori en bitbucket per tal de controlar versions, i evitar desastres. S’ha utilitzat tant en la part Front-End com en la de Back-End de l’aplicació.
- **Bitbucket:** És un servei d’allotjament basat en web, per als projectes que utilitzen el sistema de control de versions Mercurial i Git. Aquest servei és gratuït amb un número il·limitat de repositoris. S’ha utilitzat per a emmagatzemar les diferents fases del codi de Laravel i Angular.
- **Chrome:** Es tracta d’un navegador web creat per Google. S’ha utilitzat per provar l’aplicació web de l’escola, i per cercar informació necessària per a realitzar el projecte.



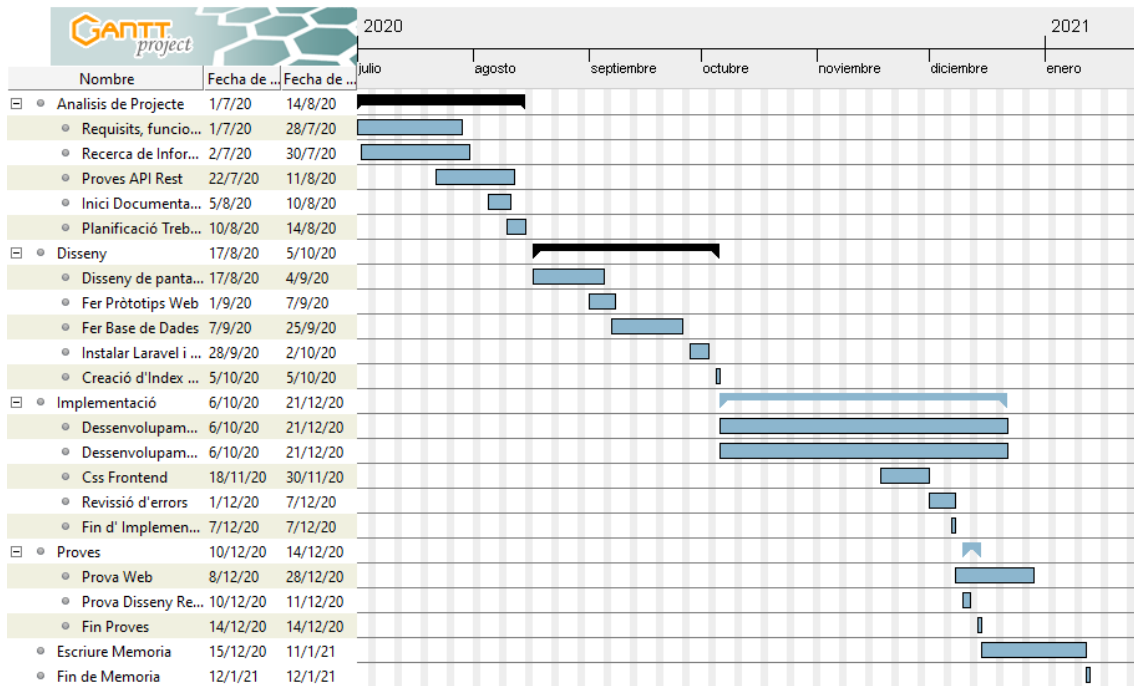
- **Passport:** És un mòdul oficial de Laravel que ens ajuda a implementar servicis d'autenticació fent ús del protocol OAuth2. S'ha utilitzat per autenticar usuaris a l'API de Laravel.
- **Bootstrap:** Bootstrap és una biblioteca multiplataforma o conjunt d'eines de codi obert per a facilitar el disseny d'aplicacions web. S'ha utilitzat en el Front-End.
- **Angular Material:** Angular Material és una llibreria d'estils com Bootstrap. S'ha utilitzat per al Front-End.

9.2. Maquinari

Per a la realització d'aquest projecte, pel que fa al maquinari, s'han utilitzat els següents dispositius:

MAQUINARI	
Ordenador	
Placa Base	MSI Z270 Gaming M5 (M5-7A78)
RAM	Corsair 4x8Gb 3200MHz DDR4
CPU	Intel i7-7700K CPU 4.20Ghz
SSD	860 EVO M.2 500Gb
HDD	Toshiba 2Gb
Pantalles auxiliars	
	Dell 27"
	Acer 24"
Ordenador Portàtil	Msi G62 Apache Pro

10. PLANIFICACIÓ



IL·LUSTRACIÓ 2 DIAGRAMA DE GANTT

Al tractar-se d'un projecte programari, el projecte s'ha dividit en 4 fases. La fase d'anàlisi, on s'exposen els requisits necessaris per fer aquest projecte, diferents proves, aprenentatge dels llenguatges que anem a utilitzar, inici de la memòria.

La fase de disseny on ens mostra una breu idea sobre com quedarà el projecte una vegada implementat, es realitzaran diferents prototips, també es dissenyarà el Back-End i el Front-End.

La fase d'implementació i proves, sens dubte la més llarga, és on codifiquem el que hem dissenyat, a banda de les revisions d'errors i validacions per a garantir que tot el codi funcione de forma exigida.

I per últim ens queda redactar la memòria.

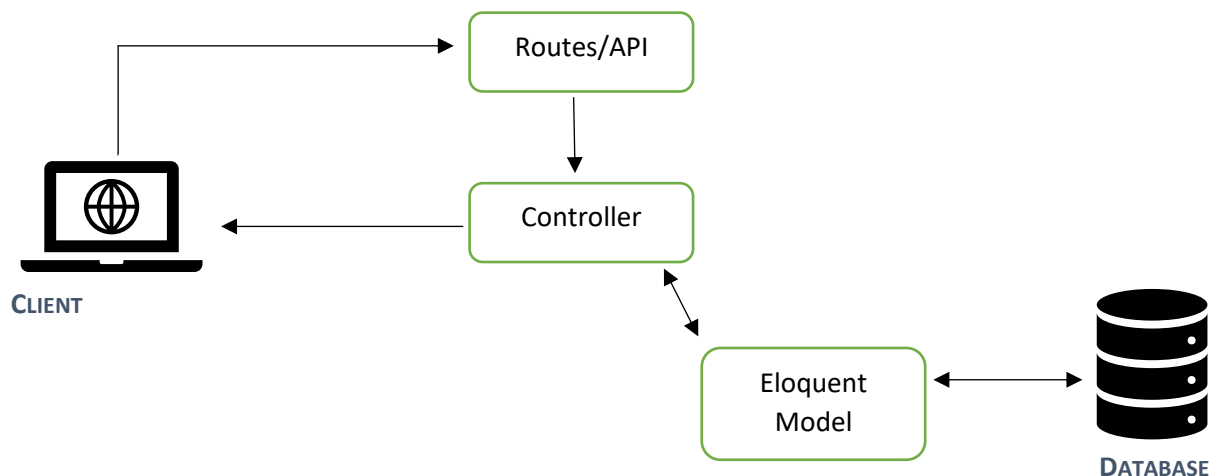
11. ANÀLISI I DISSENY

En aquest apartat, es centrarem a explicar amb més profunditat l'arquitectura del projecte, presentada en el plantejament inicial. Primer que res s'especificarà l'estructura de l'API REST, després el model de la base de dades i per finalitzar el disseny de l'aplicació web.

11.1. Estructura de l'API REST

Per al desenvolupament del Back-End s'usarà la tecnologia del Framework de Laravel. Aquest potent Framework creat per **Taylor Otwell** en l'any 2011, facilita als desenvolupadors, la creació d'Apis REST ful de forma ràpida i amb un codi net, elegant i ordenat. És a dir, ens permet crear tant aplicacions simples com aplicacions complexes, en funció del que es desitge.

Aquest Framework PHP usa el paradigma POO (Programació Orientada a Objectes). Cal destacar la seua professionalitat i velocitat de desenvolupament. Aquest Framework utilitza una variant del patró MVC (Model Vista Controlador), a continuació, es pot veure com funciona Laravel amb més detall:



IL·LUSTRACIÓ 3. PATRÓ DE DISSENY MVC DE LARAVEL

El Client fa una crida http a Laravel, que la rep el model de routes, que alhora si passa la validació, passa al controller que mitjançant el model Eloquent consulta la base de

dades i retorna la informació al control·ler per retorna l'informació al client en format JSON.

A l'aplicació de l'escola s'ha decidit utilitzar diferents mòduls per separar cada operació de l'API. Aquesta estructura multicapa ens permet augmentar la seguretat i la modularitat del projecte, conseguint així que si en algun moment del futur és necessari canviar codi, els canvis afecten la mínima quantitat possible.

Les capes en què està basada l'API Rest són les següents:

- Capa de Seguretat o Middleware (Routes/API)
- Capa de Lògica de dades de l'api (Controller).
- Capa de Base de Dades(Eloquent Model).

11.1.1. Capa de Seguretat o Middleware

Aquesta és la primera capa que és l'encarregada de rebre totes les peticions del client. Aquesta capa gestiona si les capçaleres vinents de la petició del client contenen el token d'autenticació per poder donar o negar accés als mòduls.

Aquest token, és una capa de seguretat i consisteix en una sèrie de caràcters creats a partir del nom d'usuari, la data de creació del token, la data d'expiració del token i una paraula secreta (quan més complexa millor) emmagatzemat al sistema. Aquest token deu ser únic per a cada usuari indistintament del temps.

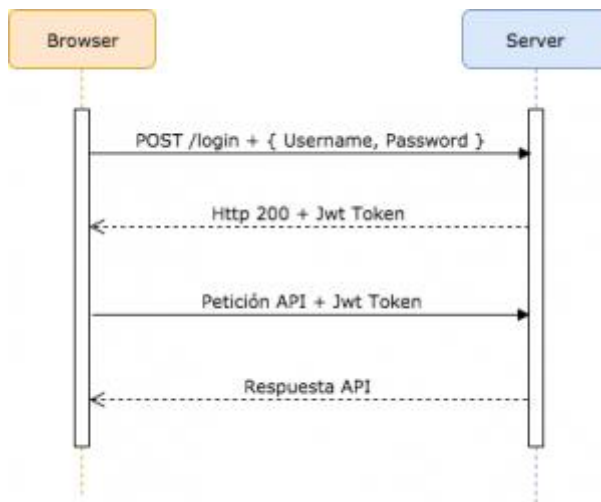
El funcionament és el següent: quan un usuari inicia sessió per primera vegada i envia les dades des del client, l'api rep les dades i si aquestes són correctes, automàticament l'api genera un token d'access únic i li retorna al client. Una vegada el client ha obtingut el token, ja pot fer crides a l'api, ja que en les peticions del client estarà present la capçalera de token.

Un token vàlid per part del client, equival a passar a la següent capa, a la capa lògica de dades.

La capa d'autenticació, ve donada pel mòdul oficial de Laravel, i hi ha que instal·lar el paquet anomenat Passport, que serà l'encarregat de servicis d'autenticació basats en el protocol OAuth2.

A continuació podem veure un exemple d'autenticació, basada en token.

Aplicació multiplataforma per a la gestió d'una escola musical



IL·LUSTRACIÓ 4 EXEMPLE D'AUTENTICACIÓ BASADA EN TOKEN

FONT: 1 [HTTPS://WWW.ADICTOSALTRABAJO.COM/WP-CONTENT/UPLOADS/2017/08/DIAGRAMA-JWT-300x249.PNG](https://www.adictosaltrabajo.com/wp-content/uploads/2017/08/DIAGRAMA-JWT-300x249.PNG)

També cal destacar que l'aplicació de l'escola a banda del middleware d'autenticació, conté un altre middleware de rols. Aquest middleware s'encarrega de donar accés a rutes, sols a l'usuari que continga aquest rol.

A continuació es mostra un fragment de codi extret de l'api on es pot observar les rutes protegides pels diferents middlewares.

```
Route::group(['middleware' => 'cors', 'prefix' => 'auth'], function () {
    Route::post('login', 'API\AuthController@login');
    Route::post('register', 'API\AuthController@register');
    Route::get('user', 'API\AuthController@user');
});

Route::group(['middleware' => 'auth:api'], function () {
    Route::get('/logout', 'API\AuthController@logout')->name('logout');
    Route::get('/details', 'API\AuthController@details')->name('details');
    Route::get('/admin', 'API\AdminController@index')->middleware('role:admin');
});
```

IL·LUSTRACIÓ 5 FRAGMENT DE CODI EXTRET DE RUTES/API.PHP

11.1.2. Capa de Lògica de Dades

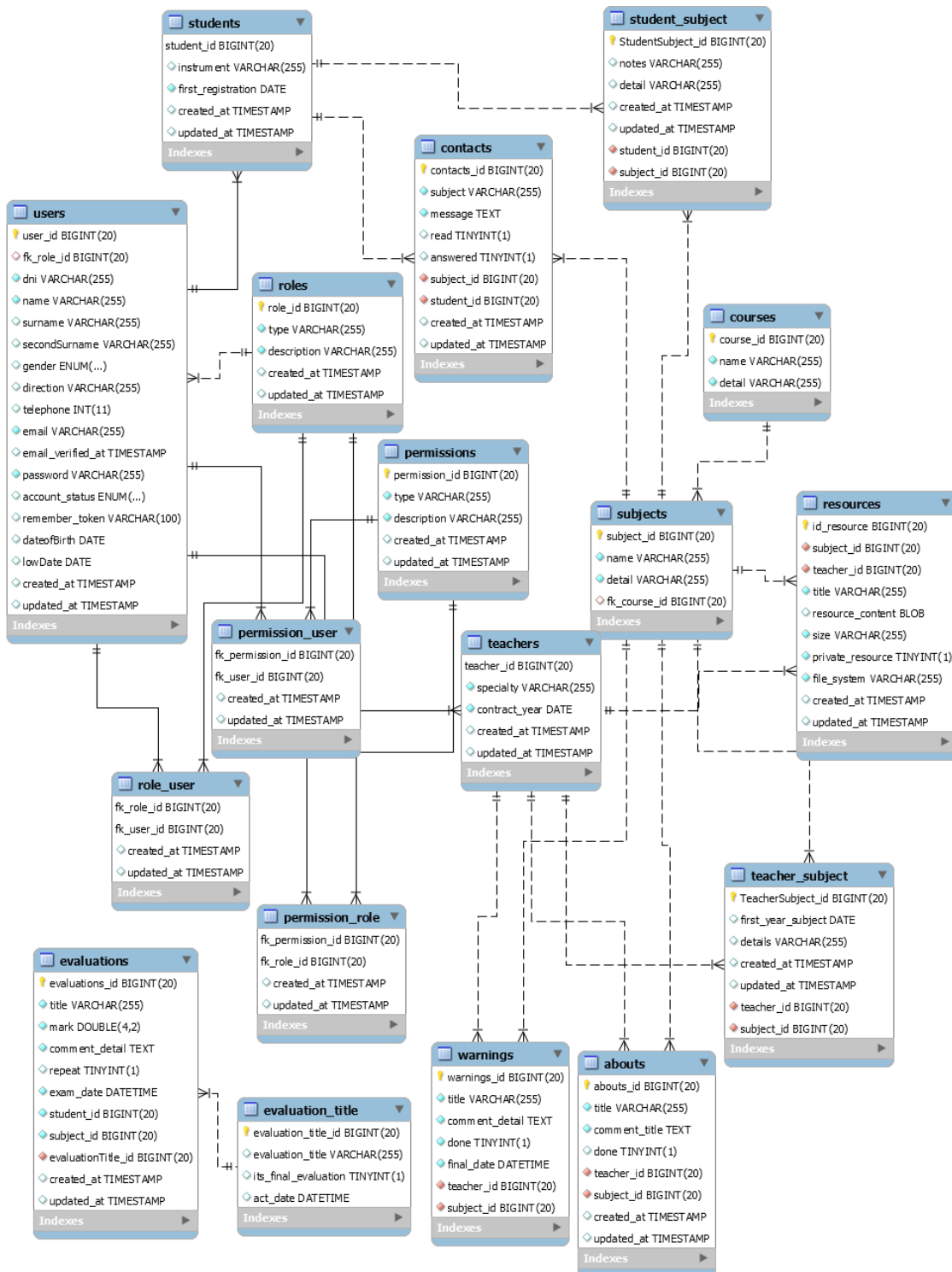
En aquesta capa es trobarà el gruix de la lògica referent a la gestió de les peticions realitzades pels clients i les respostes que se'ls enviaran.

Aquesta capa està a la vegada dividida en tres submòduls, aquests tres submòduls en Laravel s'entén com el Model, el Controlador i les Migracions. El Model és un sistema de mapeig de dades relacionats que Laravel anomena Eloquent ORM i que és l'encarregat de facilitar la creació de models. El Controlador conté la lògica de l'aplicació i permet organitzar el codi en classes sense haver d'escriure tot en la ruta. Les Migracions en Laravel, ens permeten crear taules, establir relacions, modificar-les i per supòst eliminar-les mitjançant la consola d'ordres. Tots aquests grups treballen conjuntament per respondre a les peticions del client. Els models que participen en la capa lògica a l'aplicació de l'escola són:

- Model de Auth: És l'encarregat de comprovar l'apartat d'autenticació de la capa de seguretat, s'encarrega de registrar usuaris, inici d'usuaris i la sortida d'usuaris de l'aplicació, tot relacionat amb la creació o destrucció del token.
- Model d' Informació: El Model d' Informació és l'encarregat de gestionar noves informacions sobre el model d' assignatures. Entre les funcions que realitza està la de llistar tots les informacions d'una assignatura determinada, i crear noves informacions a una certa assignatura.
- Model de Contacte: Aquest model s'encarrega de comunicar el model d'estudiants amb el model de Professors. També llista cada petició que el model estudiant haja fet i la mostra.
- Model de Curs: Aquest model s'encarrega de gestionar els cursos de les assignatures, i llistar els cursos.
- Model d'Avaluacions: És el model encarregat d'avaluar al model estudiant.
- Model de Recursos: És el model encarregat d'emmagatzemar els arxius pujats des del model Professor, i també té la funció de llistar aquests fitxers emmagatzemats.
- Model d'estudiant: És el model encarregat d'obtindre totes les assignatures del model assignatura i llistar-les.
- Model de Professors: Aquest model és l'encarregat de mostrar totes les assignatures que imparteix des del model assignatura.
- Model d'usuari: És el model encarregat de mostrar la informació de l'usuari, a banda de poder crear nous usuaris.
- Model d'avís: És el model encarregat de mostrar avisos als usuaris.
- Model d'assignatura: Es tracta del model encarregat de les assignatures, llista totes les assignatures i crea noves.

11.1.3. Capa de la base de dades

A continuació es mostra el model de dades usat en l'aplicació web:



IL·LUSTRACIÓ 5 MODEL DE LA BASE DE DADES DE L'ESCOLA

Després de veure el model de dades relacional, es detallarà el significat de cada una de les taules i les relacions que les uneixen:

USERS: Aquesta taula, fa referència als usuaris del sistema, per poder accedir a la plataforma has d'estar registrat en aquesta taula.

ROLE_USER: Taula mitjana entre Usuaris i Rols, és l'encarregada d'assignar rols als estudiants ja siga: Estudiant, Professor o Administrador.

ROL: Taula encarregada de definir els Rols del sistema, en aquesta taula estaran emmagatzemats tots els rols que existeixen.

PERMISSION_ROLE: Aquesta taula, és una taula mitjana entre Permissos i Rol, és l'encarregada d'assignar els permisos als rols que hi ha al sistema.

PERMISSION: Taula encarregada de definir els permisos que tindran els rols.

STUDENT: Taula estudiant, en aquesta taula un usuari queda definit com un estudiant i se li assigna un instrument, que principalment pot ser buit.

TEACHER: Taula professor, igual que la taula Student però ara amb Teacher, un usuari queda definit com a professor i se li assigna una especialitat.

STUDENT_SUBJECT: Taula mitjana entre estudiants i assignatures, serveix per a matricular l'estudiant en assignatures.

TEACHER_SUBJECT: Taula mitjana entre professors i assignatures, serveix per a adjudicar a un professor una assignatura.

SUBJECT: Taula assignatures, on consta el nom de l'assignatura, detalls sobre l'assignatura i el curs al que pertany.

RESOURCES: Taula de recursos associats a l'assignatura, cada assignatura tindrà un o molts recursos depenent del professor, taula creada per a emmagatzemar la ruta del document pujat pel professor.

COURSES: Taula on es defineixen els cursos de l'escola.

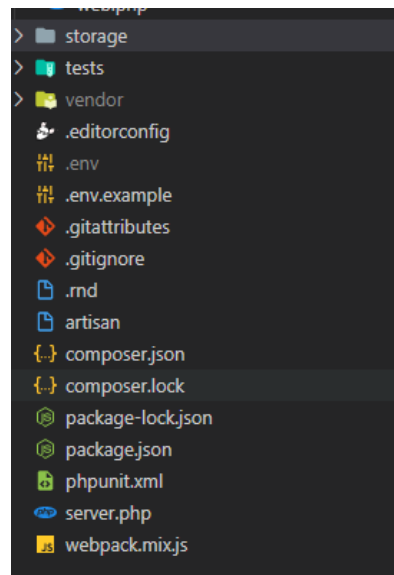
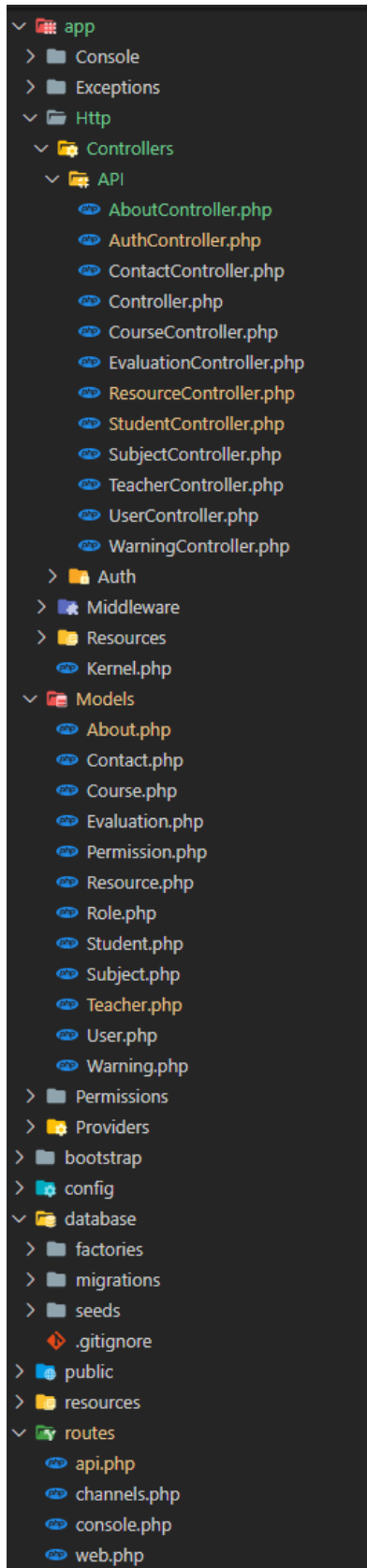
WARNINGS: Taula per gestionar els avisos als estudiants fets pel professor. Conté un boolean per canviar l'estat de privat o públic.

ABOUTS: Taula associada a una assignatura, ja que el professor de l'assignatura pot introduir com avaluarà l'assignatura o el que s'espera de l'assignatura.

EVALUATIONS: Taula no relacional en el qual s'avaluarà a l'alumne.

EVALUATIONS_TITLE: Taula per englobar tots en un acte d'evaluació.

11.1.4. Estructura de carpetes Back-End



IL·LUSTRACIÓ 6 ESTRUCTURA DE CARPETES DEL BACK-END.

11.2. Disseny de l'aplicació web

El disseny de la pàgina web s'ha vist condicionat també pel Framework seleccionat per al desenvolupament del Front-End, que com ja s'ha dit ha sigut angular.

Angular és un Framework creat per Google que es basa en aplicacions d'una sola pàgina o (SPA) en el que es basen els components. Les pàgines en angular tenen un component principal, "app.component" i a partir d'aquest component van desenvolupant-se i va creixent l'aplicació.

Els components en angular estan pensats per ser totalment reutilitzables i es basa en la idea de crear pàgines mitjançant la utilització de components més xicotets. Per exemple: es pot crear un component navbar.component i un altre component com table.component i en la app.component sols hauríem d'afegir el selector dels dos components i ja ens mostraria la barra de navegació i la taula.

També cada component creat està format per tres subcomponents: el **.html** que és el fitxer on introduïrem el codi que després vorà l'usuari, es a dir, la vista, el **.css** és l'estil que tindrà la vista del .html i per últim el **.ts** o typescript que és la lògica encarregada de donar funcionalitat al component.

Després d'explicar breument el que són els components en angular, cal tindre en compte que existeixen fitxers anomenats servicis, que són els encarregats de fer les crides http a l'api rest.

11.2.1. Estructura de Carpetes Frontend

Ara que entenem el funcionament, detallaré l'estructura utilitzada en el Front-End per a l'escola.

El punt de partida d'aquest projecte angular és app.component, a partir del qual es desenvoluparà totes les altres pàgines. S'han creat dues subcarpetes dins de la carpeta components, els components reutilitzables batejats com common i una carpeta pages per a les pàgines web.

També s'ha creat una carpeta Shared on s'engloben els distints servicis d'autenticació i gestió de token.

Una altra carpeta serà la carpeta services on s'emmagatzemaran tots els servicis de l'aplicació encarregats de consultar a l'api.

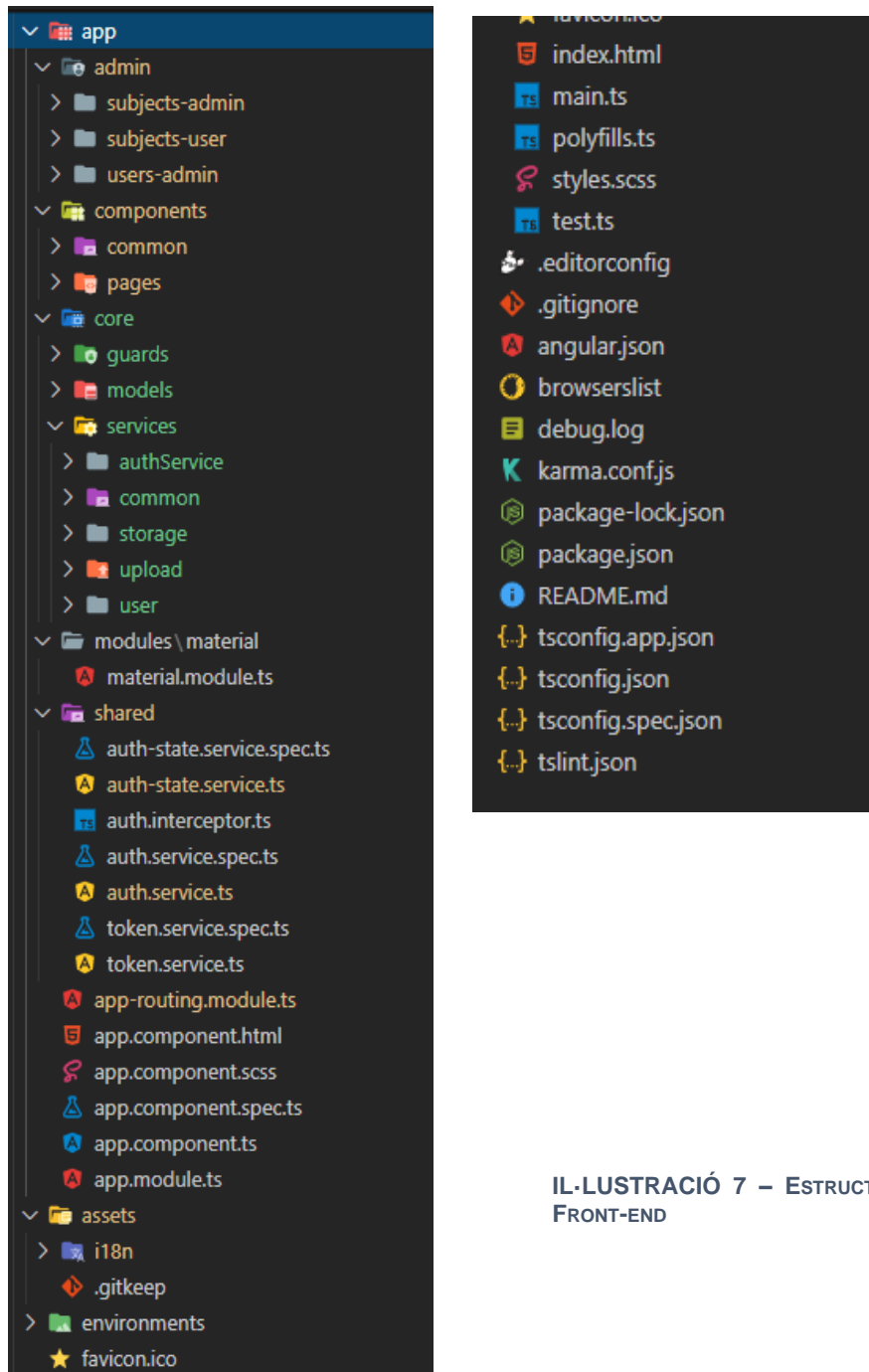
També s'ha creat una carpeta models que s'encarrega d'estructurar les dades obtingudes de peticions http.

A més a més es crea també una carpeta authGuard on es protegeixen les rutes del Routing module encarregat de la navegació de la pàgina web.

També s'ha fet un mòdul per a angular material, on introduïm els imports que ens fan falta per poder gastar els components d'angular material.

I per últim s'ha creat la internacionalització en la carpeta assets/i18n.

A continuació es mostra l'esquema de les carpetes del projecte.



IL·LUSTRACIÓ 7 – ESTRUCTURA DE CARPETES DEL FRONT-END

11.2.2. Disseny de Mockups versió escriptori

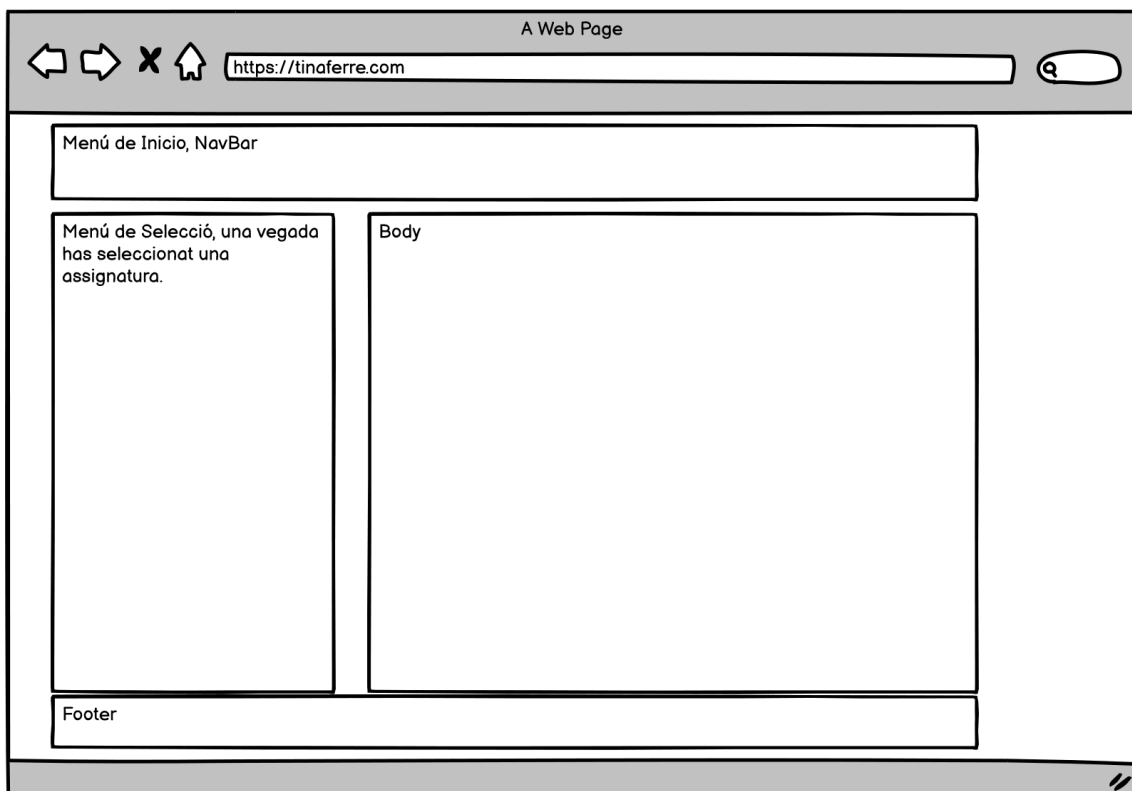
El disseny de la interfície deu ser usable, ja que l'objectiu principal és que les aplicacions siguin atractives per a l'usuari i que siguin el més intuïtives possible, per a que l'usuari no necessite informació addicional per poder navegar.

A continuació es mostrarà alguns dissenys de la interfície de l'aplicació dels distints rols que té la plataforma, per a versió web, tablet i mòbil.

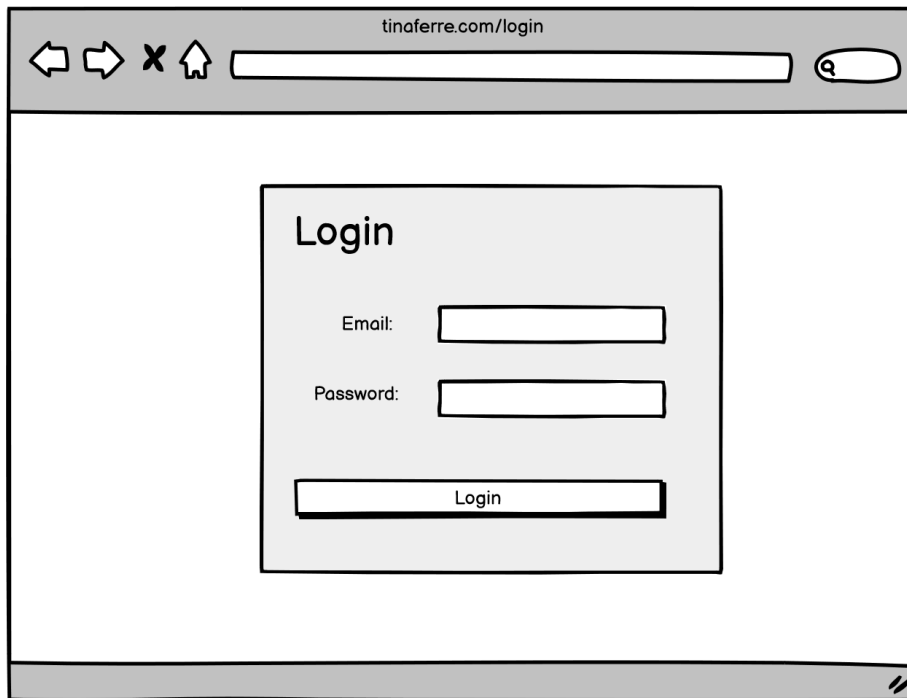
Per desenvolupar aquests prototips se seguiran les següents pautes:

- A l'hora d'implementar aquests prototips poden canviar.
- S'han obviat prototips similars entre pantalles.
- Simplicitat de prototips, ja que afavoreixen la usabilitat.

Tota l'aplicació web està basada en el següent aspecte:

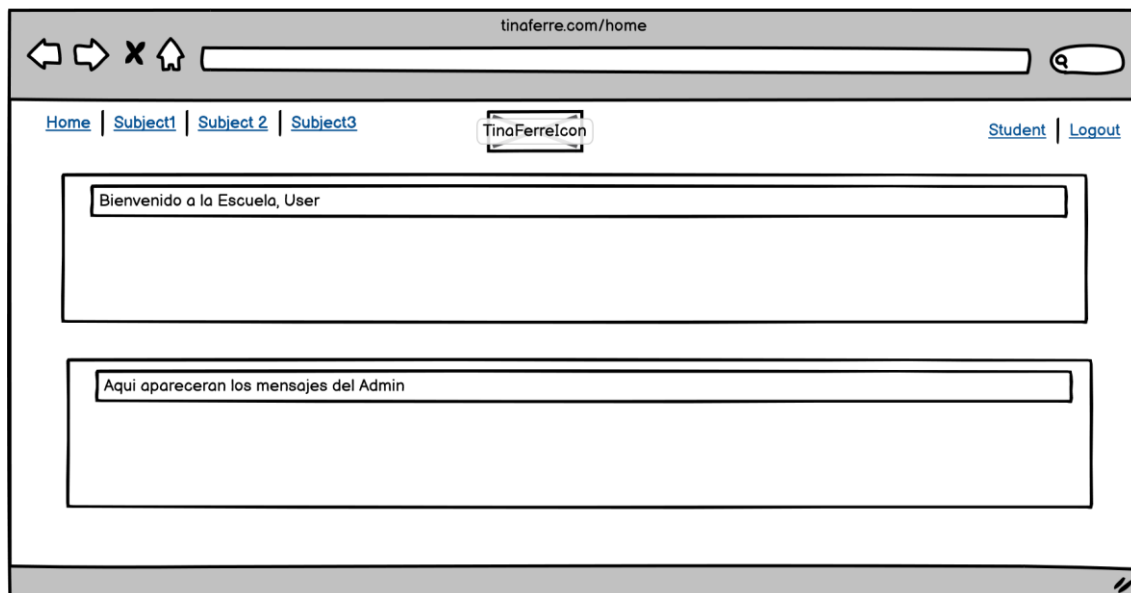


IL·LUSTRACIÓ 8 VERSIÓ ESCRIPTORI (ROL ESTUDIANT)

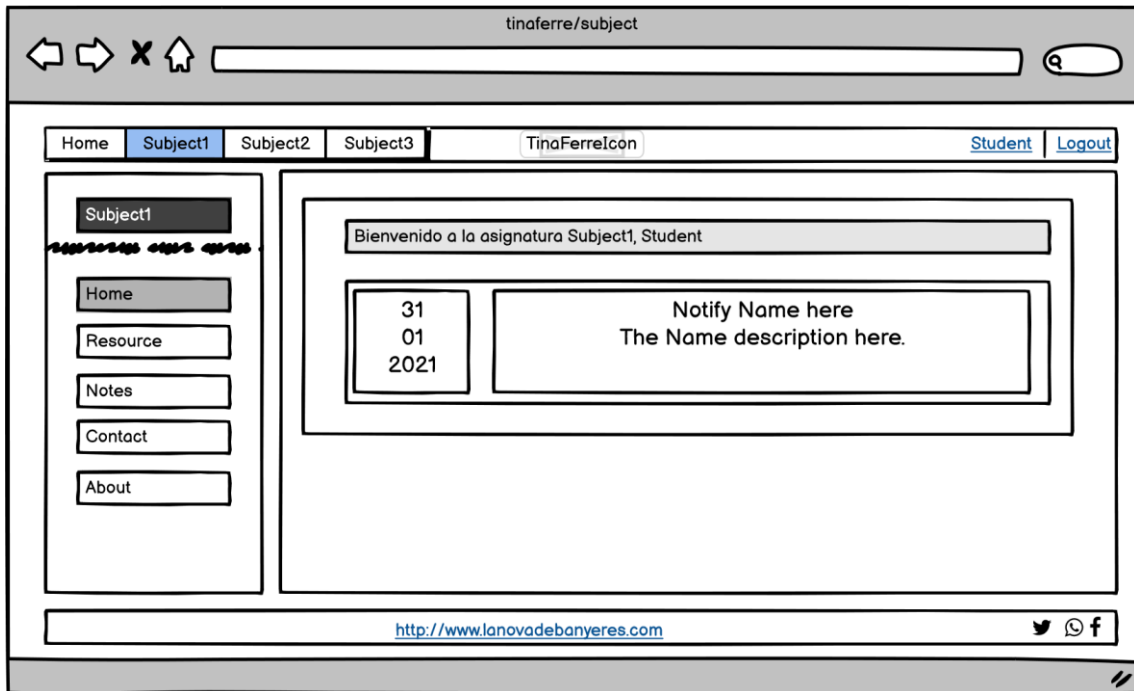


IL·LUSTRACIÓ 9 PÀGINA DE LOGIN DE L'APLICACIÓ WEB

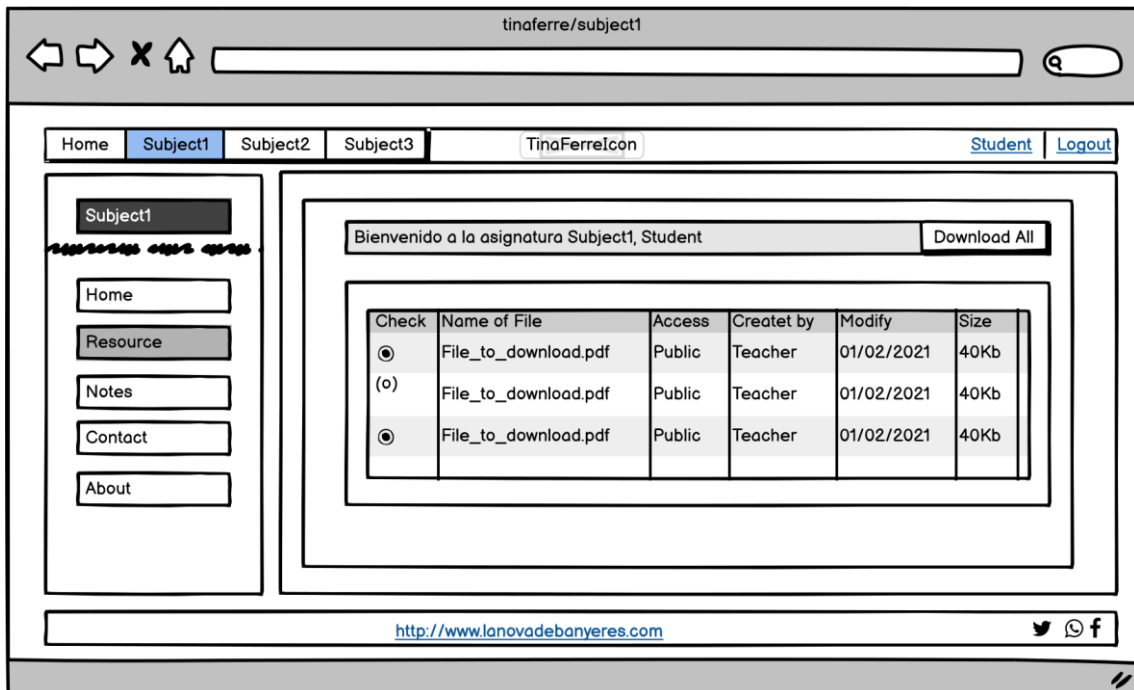
11.2.2.1. Rol Estudiant



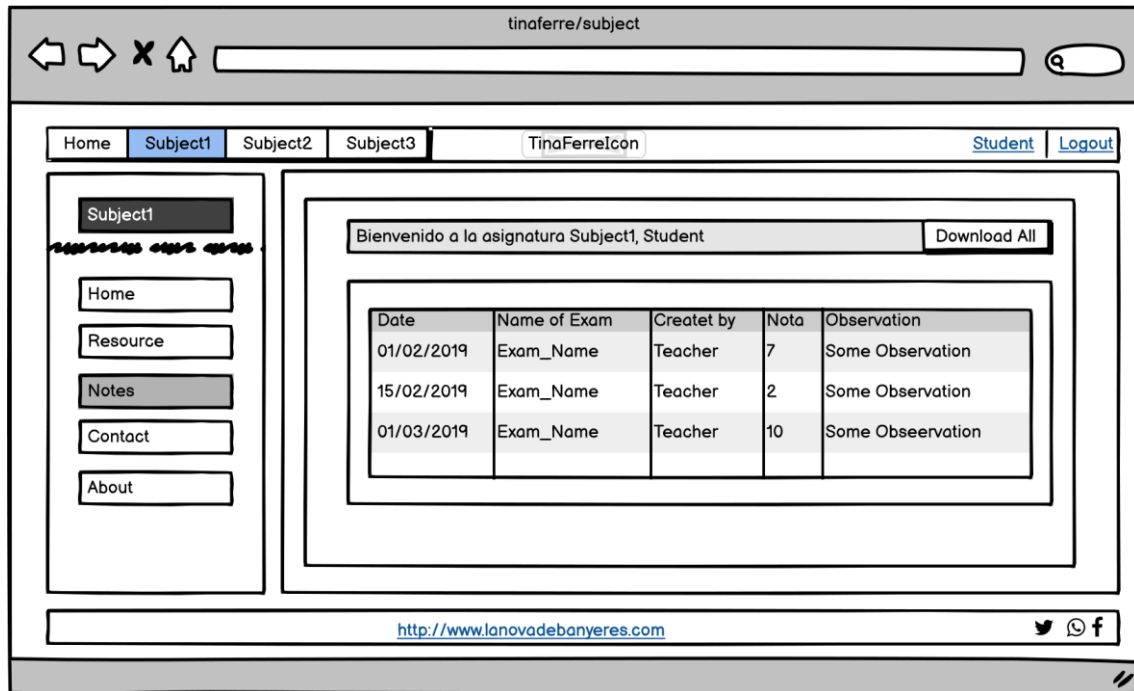
IL·LUSTRACIÓ 10 PÀGINA D'INICI DE L'APLICACIÓ WEB



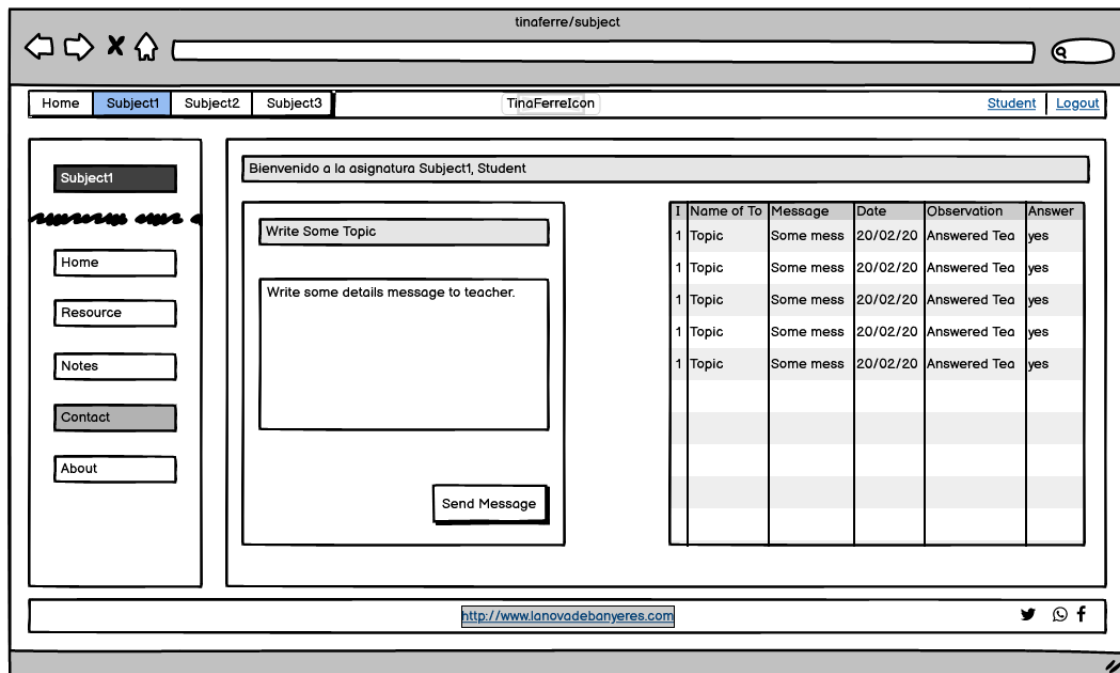
IL·LUSTRACIÓ 12 LA PÀGINA D' INICI DE L'ASSIGNATURA



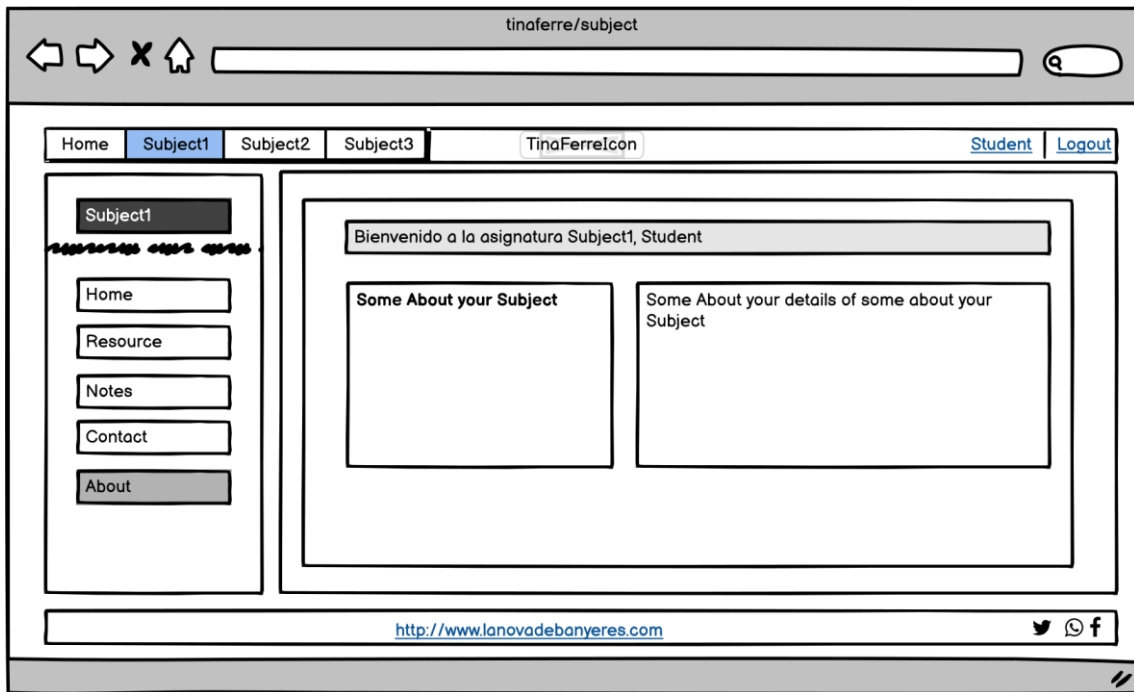
IL·LUSTRACIÓ 11 LA PÀGINA DE L'ASSIGNATURA1->RECURSOS



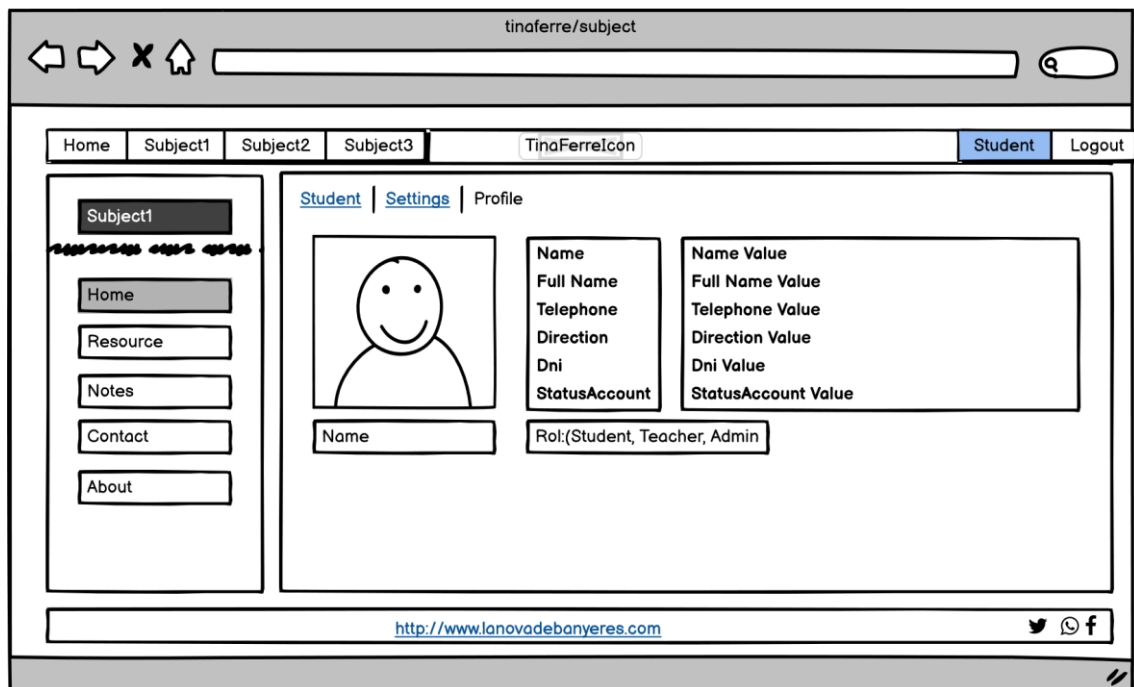
IL·LUSTRACIÓ 13 QUALIFICACIONS DE L'ASSIGNATURA 1



IL·LUSTRACIÓ 14 PÀGINA DE CONTACTE DE L'ASSIGNATURA 1

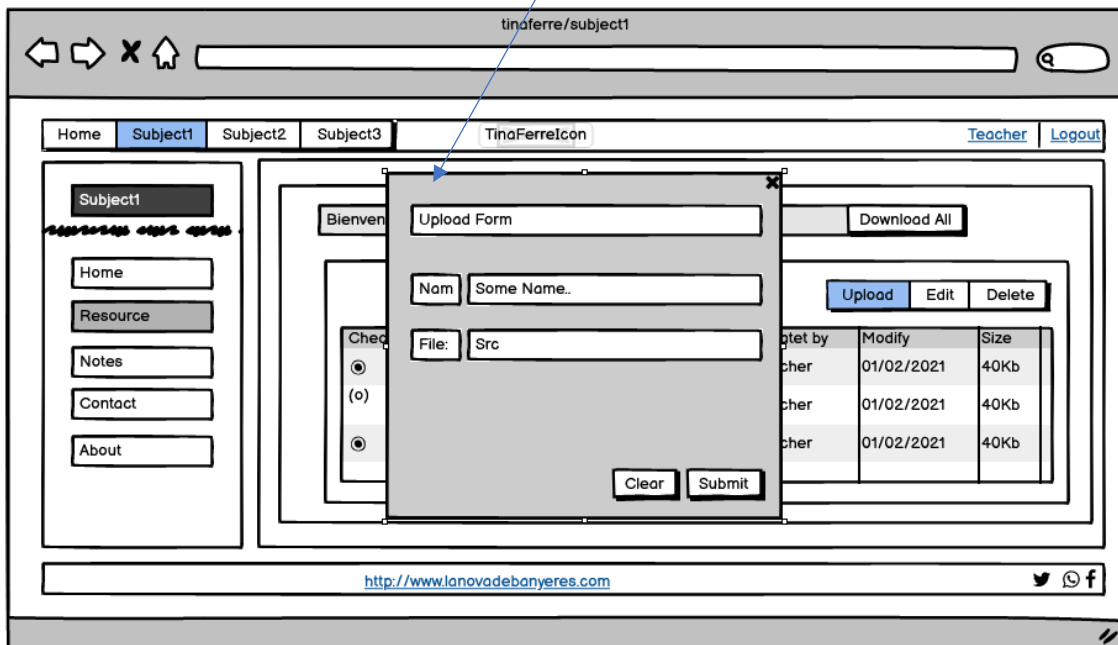
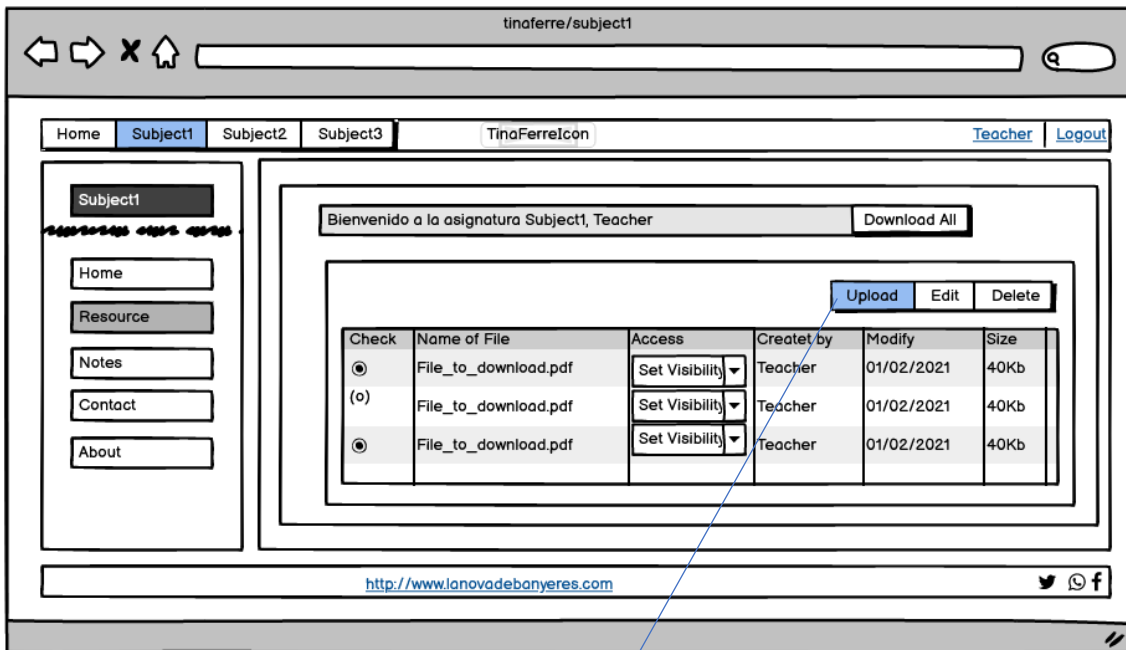


IL·LUSTRACIÓ 15 PÀGINA DE ABOUT DE L'ASSIGNATURA 1

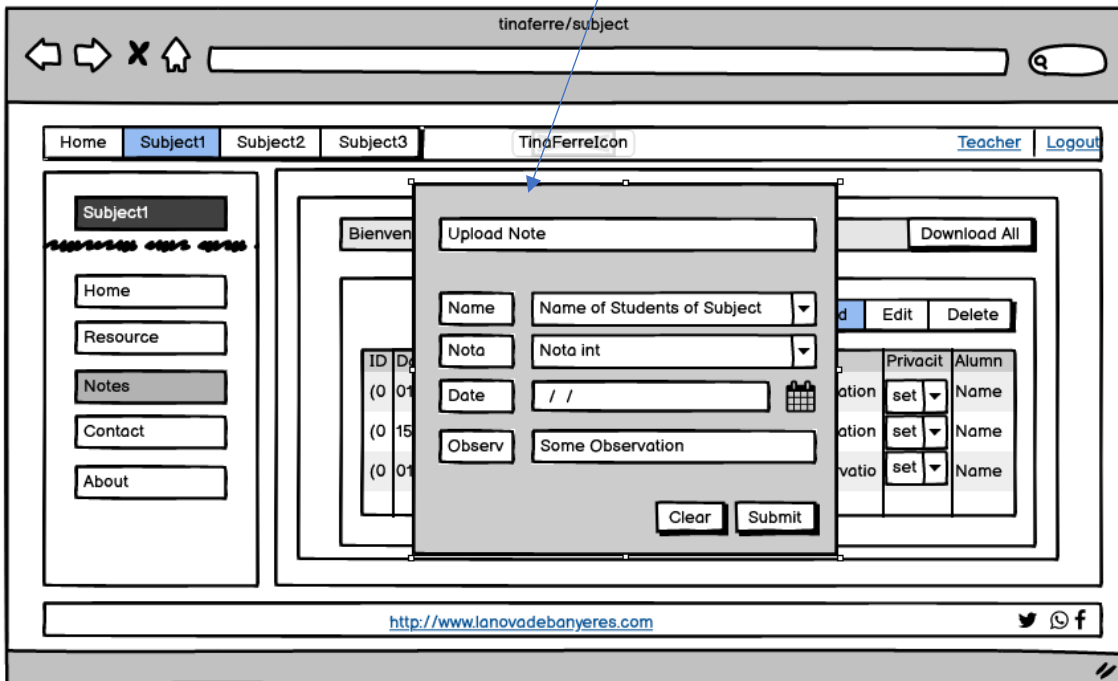
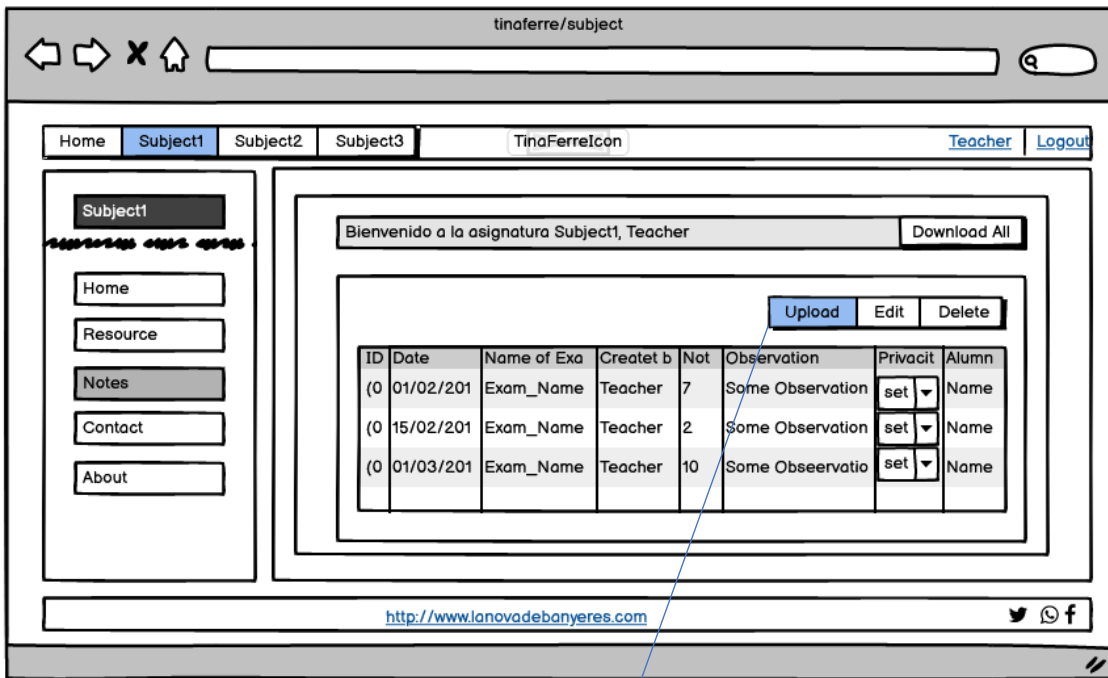


IL·LUSTRACIÓ 16 PÀGINA DE PERFIL DE L'USUARI

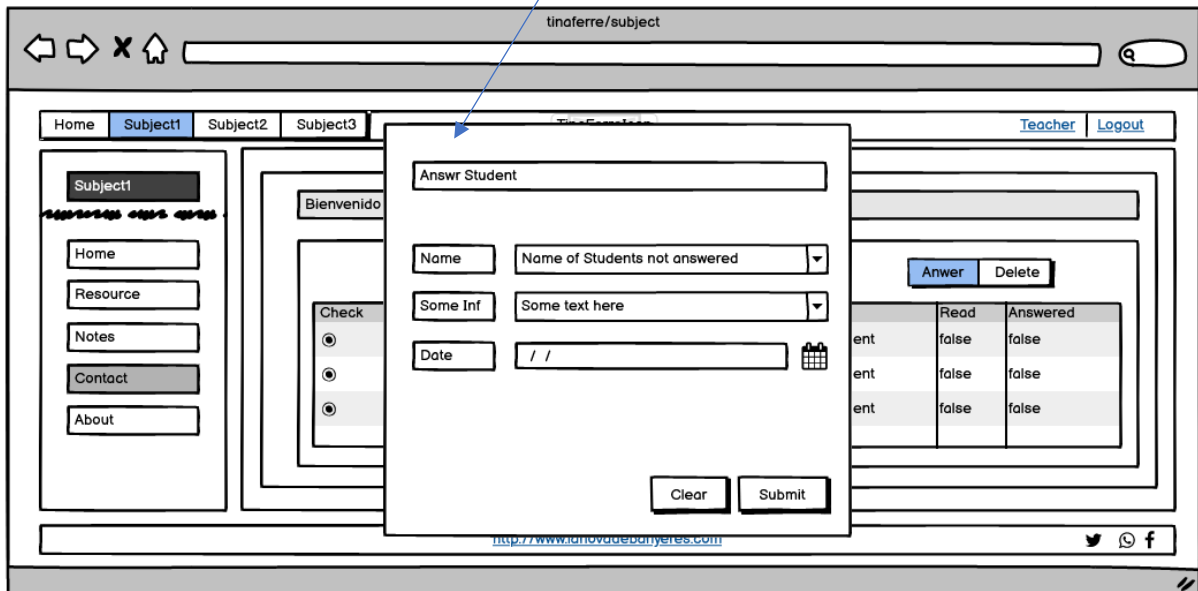
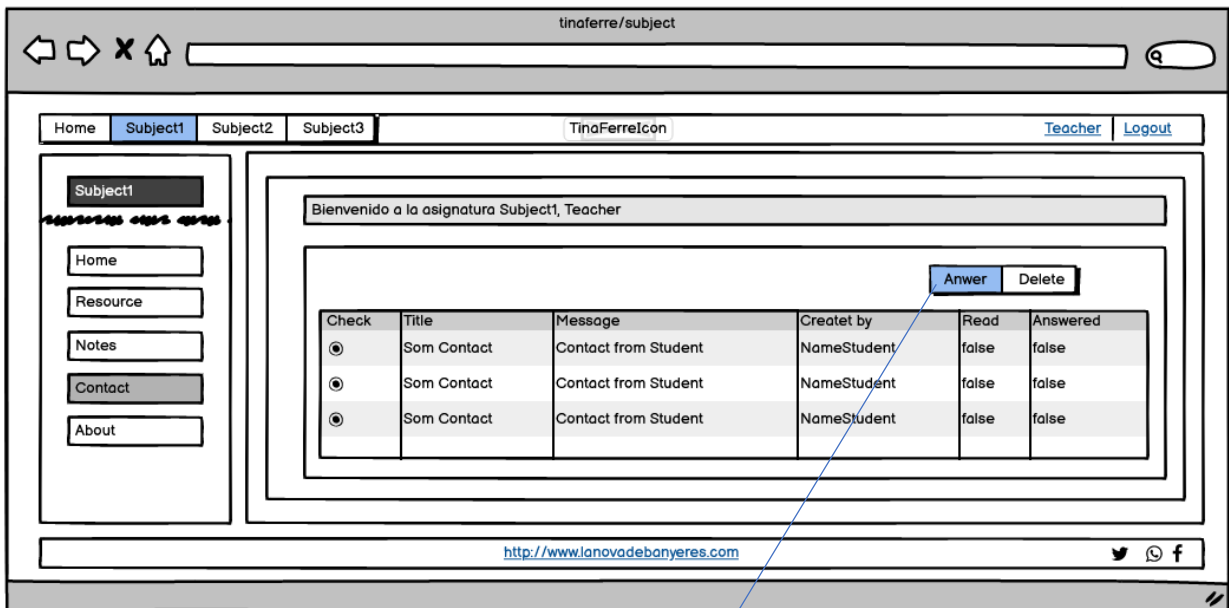
11.2.2.2. Rol Professor



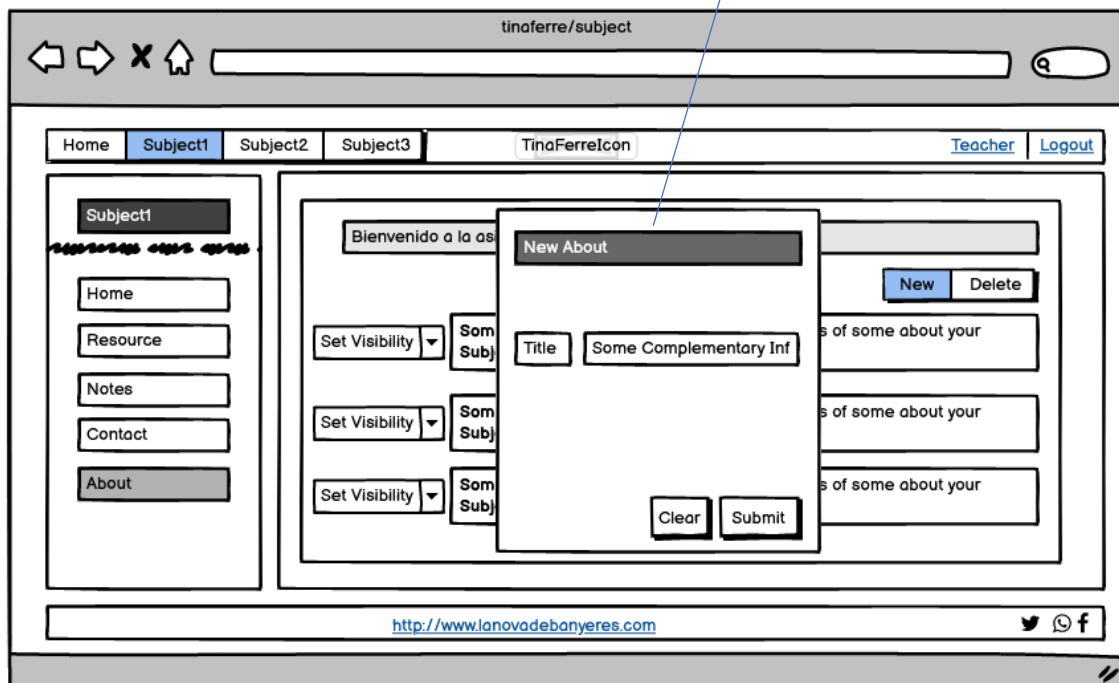
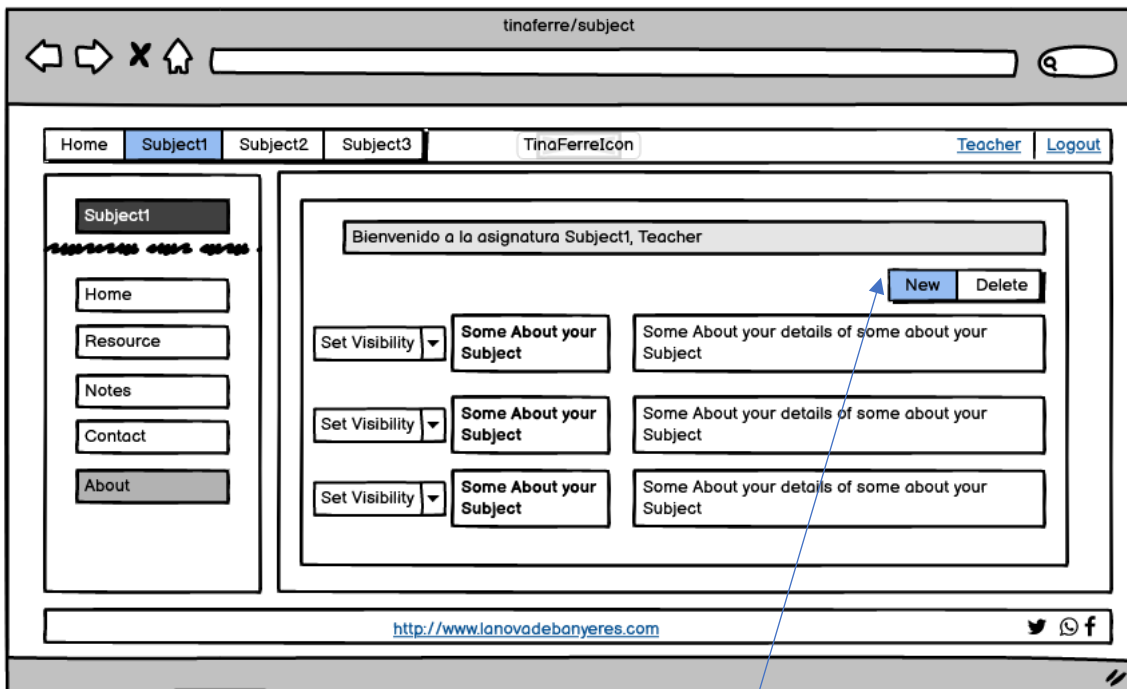
IL·LUSTRACIÓ 17 FLUX D'UN PROFESSOR PUJANT UN RECURS



IL·LUSTRACIÓ 18 FLUIX D'UN PROFESSOR PUJANT UNA NOTA

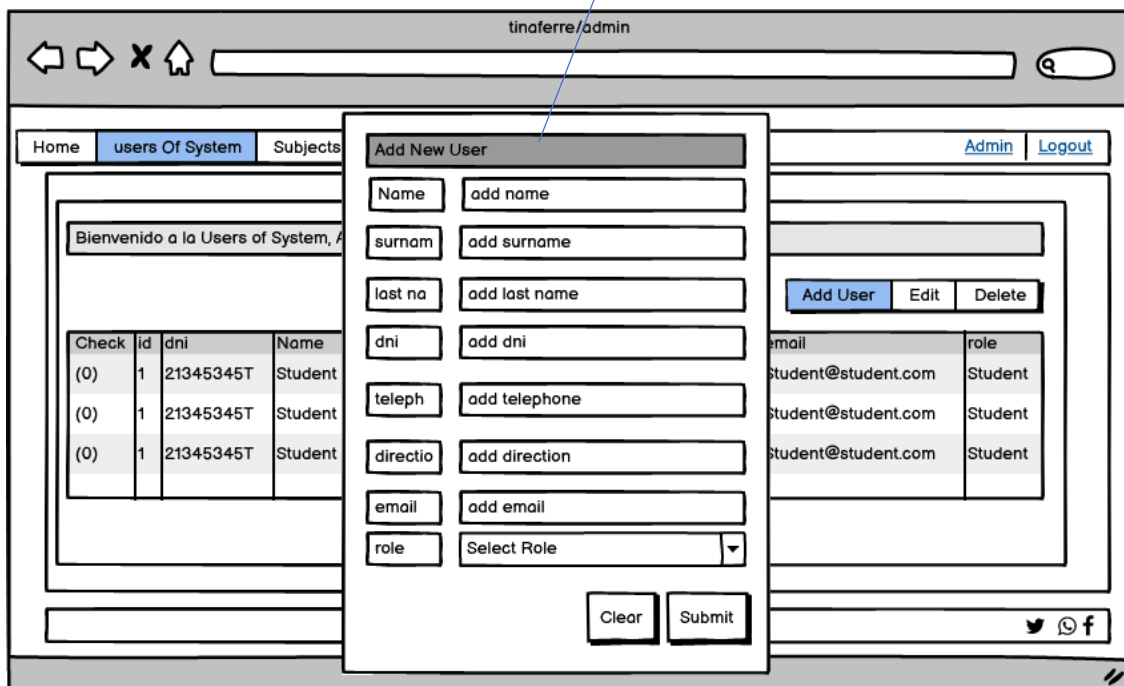
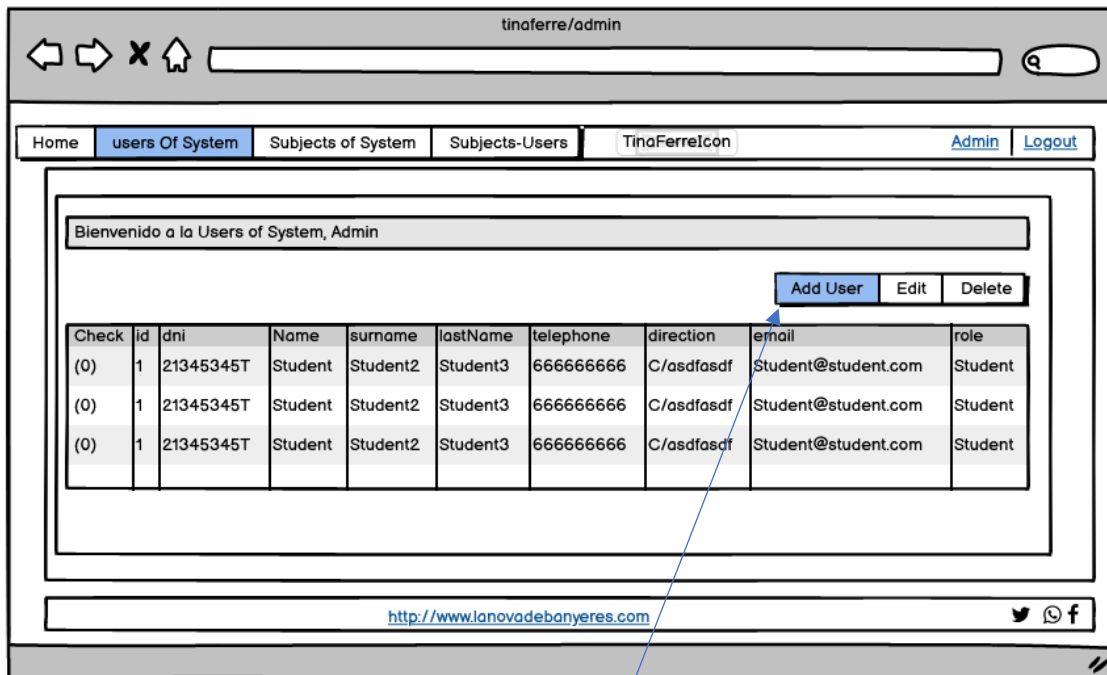


IL·LUSTRACIÓ 19 FLUX ON UN PROFESSOR RESPON A UN ESTUDIANT

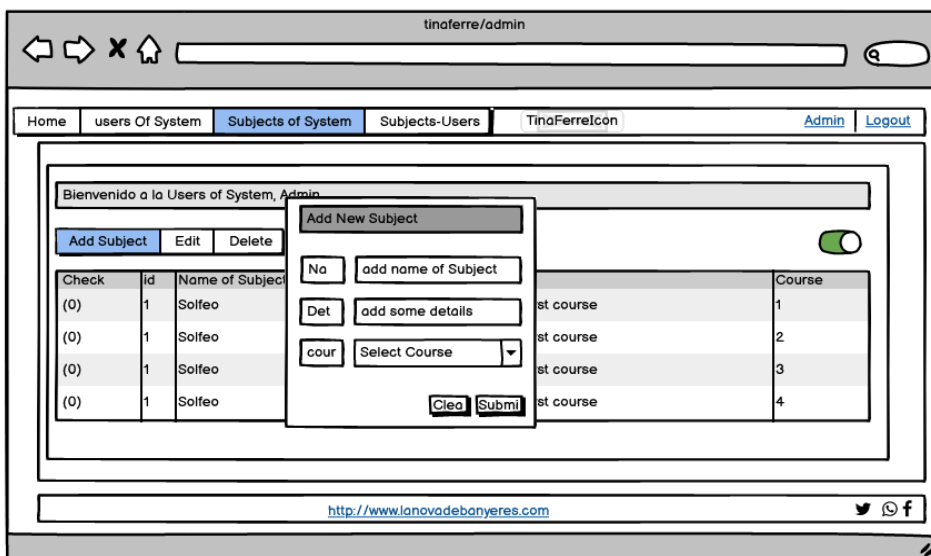
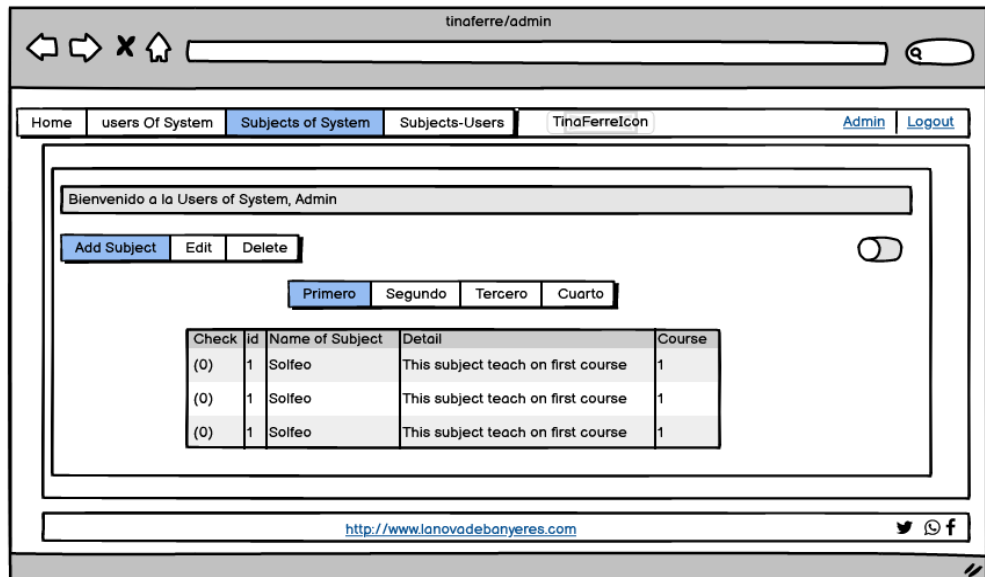
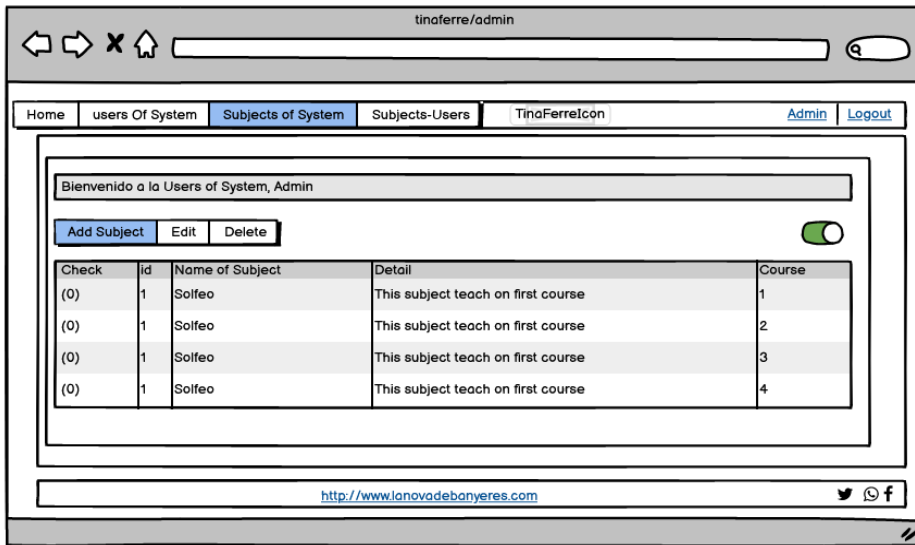


IL·LUSTRACIÓ 20 FLUIX ON UN PROFESSOR PUJA UN NOU ABOUT

11.2.2.3. Rol Administrador

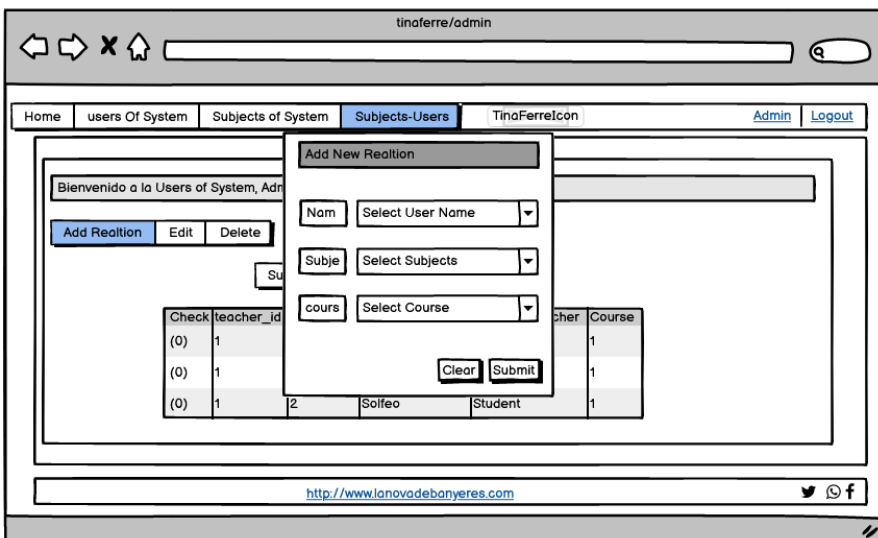
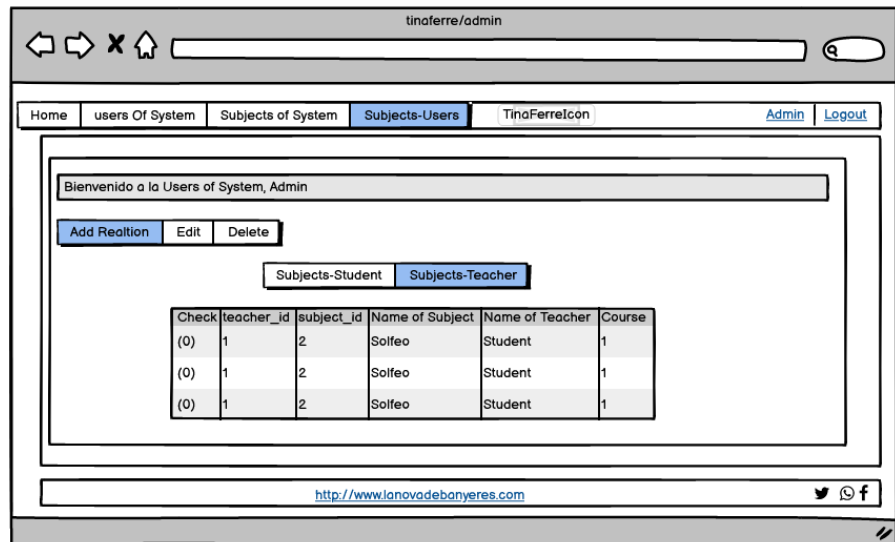
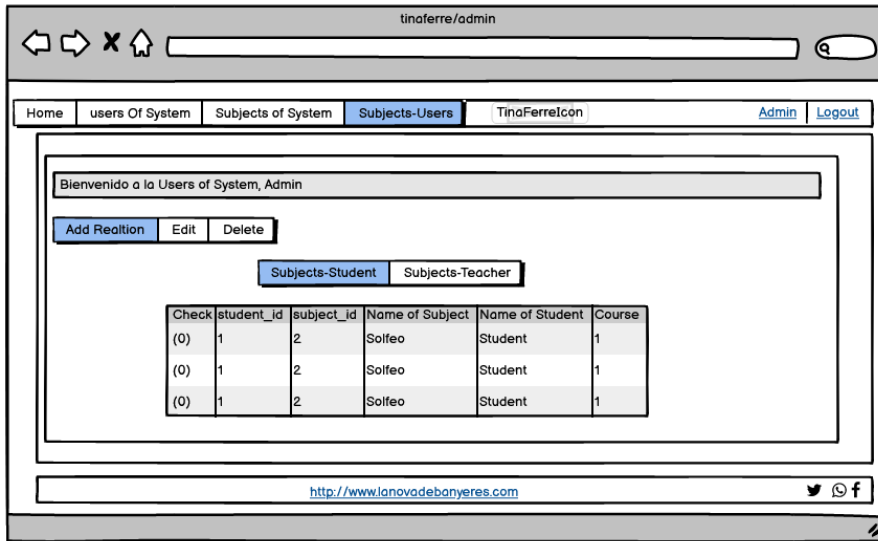


IL·LUSTRACIÓ 21 FLUIX D'ADMINISTRADOR AFEGINT UN NOU USUARI



IL·LUSTRACIÓ 22 IMATGE D'ASSIGNATURES ON ES MOSTREN (TOTES O PER CURSOS), SEGONS POLSEM EL BOTÓ TOGGLE, I PER ÚLTIM MOCKUP D'AFEGIR ASSIGNATURA

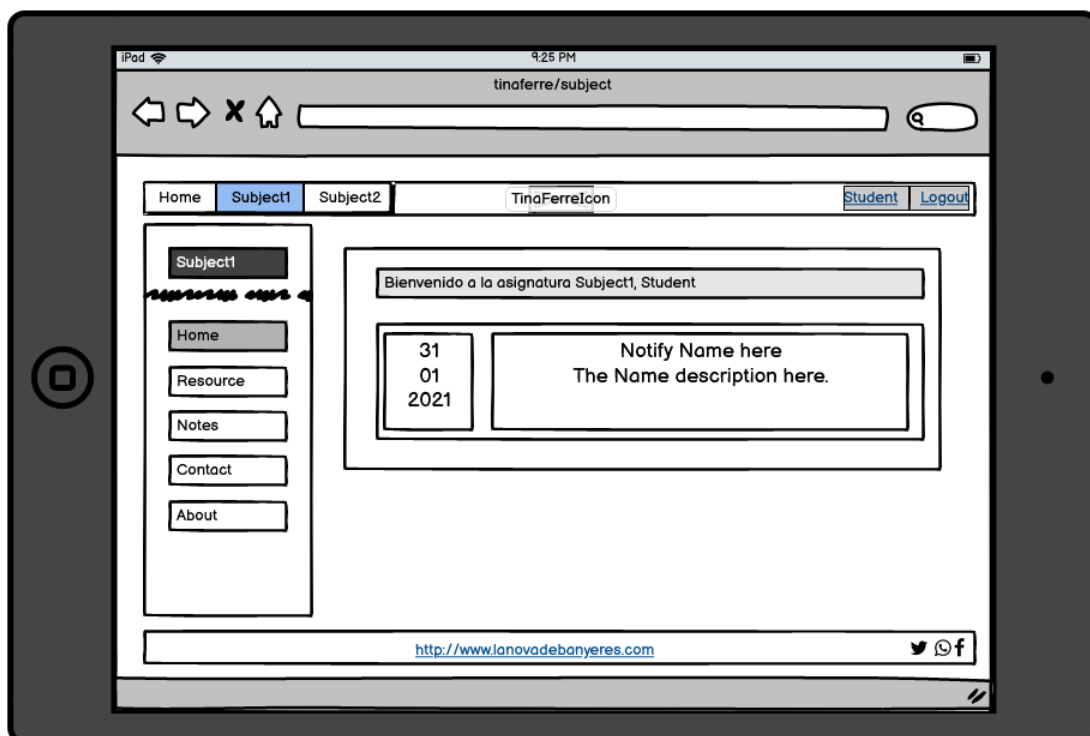
Aplicació multiplataforma per a la gestió d'una escola musical



IL·LUSTRACIÓ 23 RELACIÓ D'USUARIS-ASSIGNATURES ON, A CADA USUARI, SE LI ASSIGNA UNA ASSIGNATURA

11.2.3. Disseny de mockups versió tablet

Pàgina d' Inici d'una assignatura en l'aplicació web de la **Tablet on** Min-width: 768px / Max-width: 979px.



IL·LUSTRACIÓ 24 DISSENY D'APLICACIÓ WEB PER A TABLET

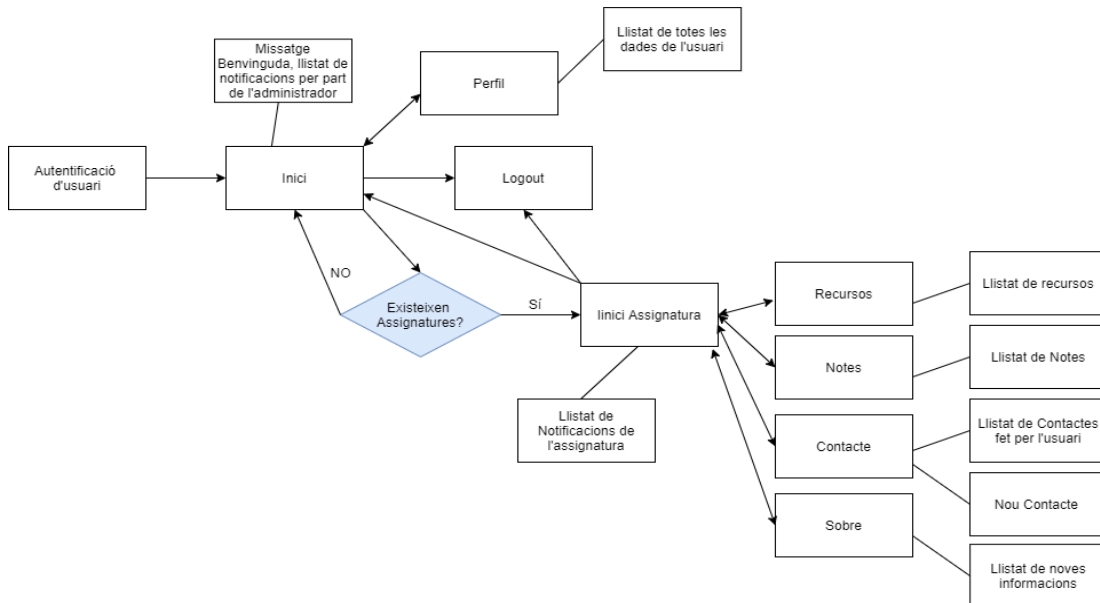
11.2.4. Disseny de mockups versió smartphone



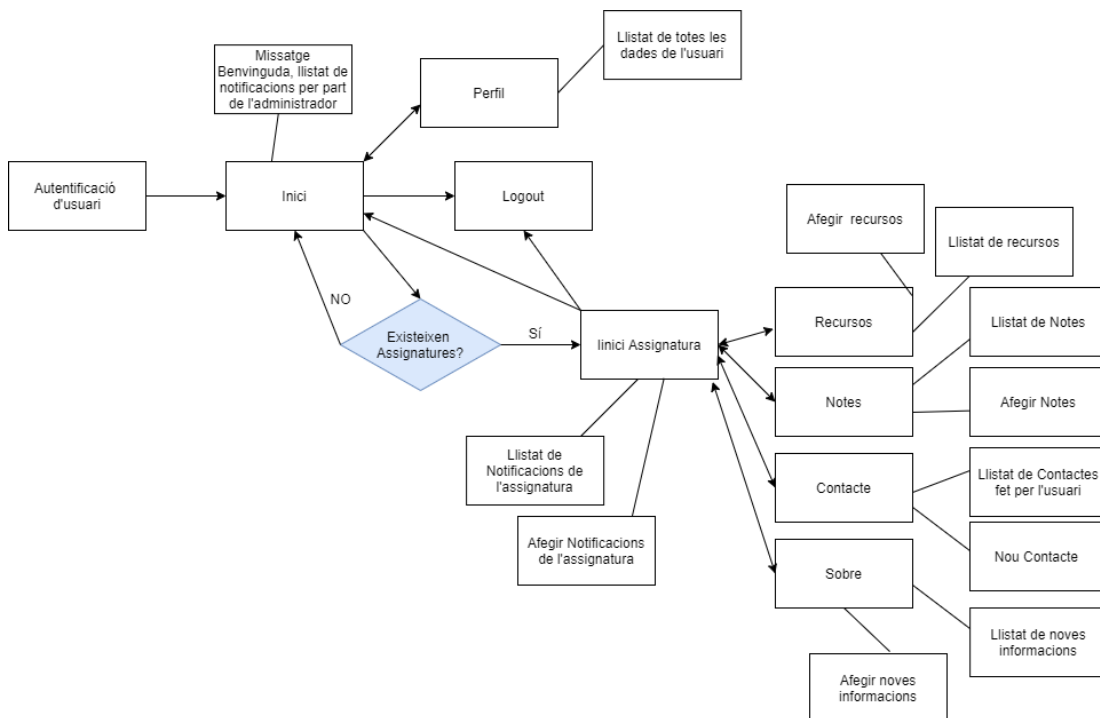
IL·LUSTRACIÓ 25 FLUX DE NAVEGACIÓ D'UN USUARI DE TIPUS ESTUDIANT. EN UN SMARTPHONE-> Max-width: 480px

11.2.5. Diagrama de funcionament

A continuació es mostren els digrames de funcionament, dels rols estudiant, professor i administratiu.

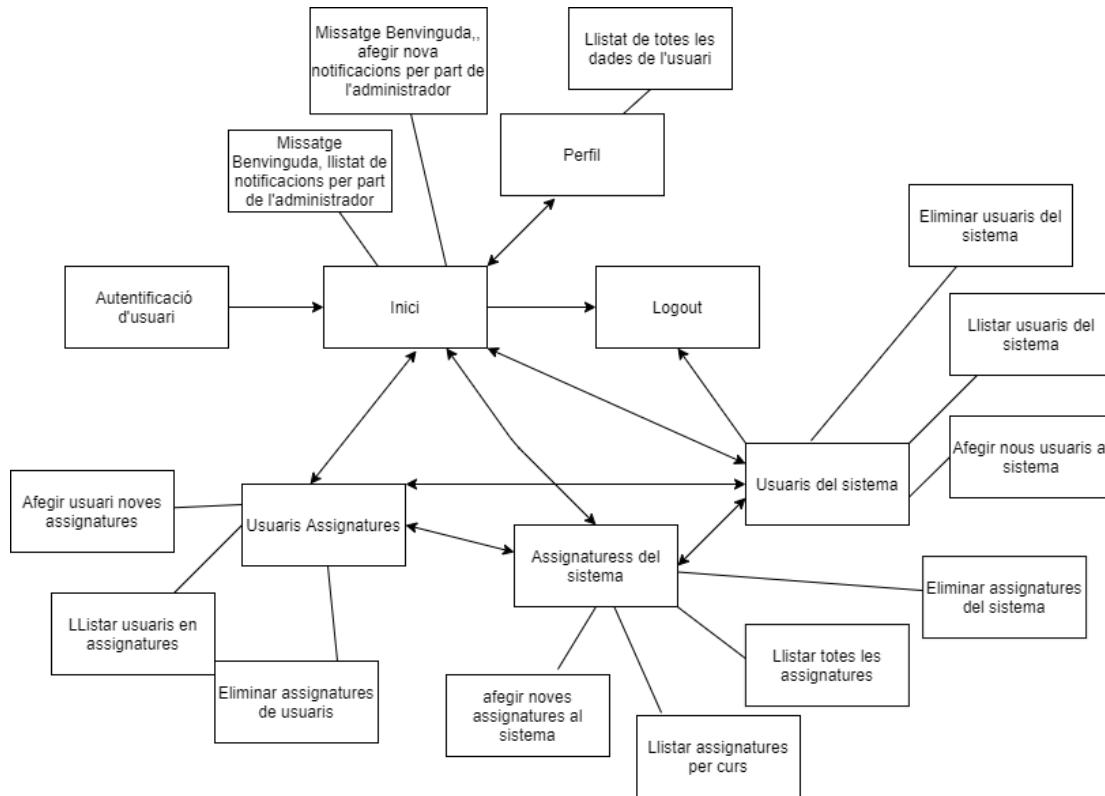


IL·LUSTRACIÓ 26 DIAGRAMA DE FUNCIONAMENT DEL ROL ESTUDIANT



IL·LUSTRACIÓ 27 DIAGRAMA DE FUNCIONAMENT DEL ROL PROFESSOR

Aplicació multiplataforma per a la gestió d'una escola musical



IL·LUSTRACIÓ 28 DIAGRAMA DE FUNCIONAMENT DEL ROL ADMINISTRATIU

12. IMPLEMENTACIÓ

En aquest apartat s'explicarà els passos que s'han realitzat per desenvolupar l'aplicació de l'escola Tina Ferre. Els apartats estaran dividits en dues parts: implementació Back-End i després implementació Front-End. S'explicarà de manera detallada com s'han realitzat algunes parts, la instal·lació de programes addicionals, llibreries necessàries, així com els punts clau del desenvolupament.

12.1. Implementació del Back-End

Abans d'entrar amb detall, sobre el desenvolupament del Back-End, vaig a esmentar els passos per poder instal·lar les ferramentes que s'utilitzaran durant el transcurs del desenvolupament del Back-End.

12.1.1. Instal·lació Back-End

Per a instal·lar Laravel, primer que tot necessitem tindre instal·lat PHP, si no el tenim ens descarreguem XAMP i executem l'instal·lador. Elegim on voldrem la instal·lació i després l'instal·lador associarà PHP a la variable d'entorn (PATH).

Una vegada finalitzat l'instal·lador per comprovar que funciona PHP correctament, obrim el símbol del sistema o consola i teclegem `$php -v`, si ens mostra la versió s'ha instal·lat correctament. Si no, a vegades hauràs d'introduir el PATH manualment.

Una vegada ja tenim instal·lat PHP, necessitem instal·lar Composer. Composer s'encarrega d'instal·lar les llibreries que el nostre projecte necessitarà i també fa la funció de mantindre actualitzat totes les llibreries. Per a instal·lar Composer cal descarregar-lo des de la seua pàgina oficial i executar l'instal·lador.

Una vegada instal·lat crearem el nou projecte Laravel. Per fer-ho, des del terminal teclegem:

```
Composer global require "laravel/installer"
```

```
Laravel new backend
```

Una vegada s'haja creat el projecte l'obrim en el nostre IDE de desenvolupament i modifiquem el fitxer.env i teclegem:

```
Php artisan key:generate
```

```
Php artisan serve
```

Ara obrim el navegador i entrem a <http://backend.dev> i visualitzarem una pàgina pareguda a açò:

Laravel

DOCUMENTATION

LARACASTS

NEWS

FORGE

GITHUB

IL·LUSTRACIÓ 29- PÀGINA D'INICI DEL PROJECTE CREAT EN LARAVEL

Una vegada comprovem que el projecte funciona, dins de la carpeta instal·lem les dependències amb el codi:

Composer install

Creem una base de dades mysql desde Xampp.

I connectem Laravel amb la base de dades introduint en el fitxer .env de Laravel la base de dades creada en mysql, a banda d' introduir l'usuari i contrasenya.

Per a mostrar una llista de tots els recursos que hi ha disponible utilitzem aquest codi: **php artisan list**, i per obtindre ajuda: **php artisan help** .

En la carpeta /config estan els fitxers de configuracions que contenen les instruccions necessàries per a carregar i configurar el Framework. El fitxer config/data base.php conté variables per a configurar la base de dades a la que es va a connectar.

D'altra banda, totes les peticions del client arriben a públic/index.php i després són interpretades pel fitxer de rutes app/Http/routes.php on es defineixen els Controllers que s'encarreguen de processar la petició.

Amb **php artisan migrate:install** instal·lem les migracions, per tal de dur a terme el control de versions de la base de dades, tant per a crear les taules com per a eliminar-les. Per a carregar dades de prova ràpidament a la taula es poden usar seeders: **php artisan db:seed**.

Una vegada comprovem que tenim accés a la base de dades, creem els models amb les migracions amb el comandament:

Php artisan make:Model About -m

Este codi ens haurà creat una migració i un model de nom About

Ara creem el controlador per a aquest model amb:

Php artisan make:controller AboutController

I aquest codi ens haurà creat l'imprescindible per poder començar a treballar amb Laravel.

Una vegada ja tenim Laravel funcionant i sabem crear models, controladors i migracions, ens falta l'autenticació. Necessitem instal·lar el paquet Laravel Passport i configurar-lo. Des del terminal teclegem:

Composer require laravel/passport

Ara necessitem executar migracions perquè el paquet Laravel utilitza algunes taules en la base de dades, i per aquesta raó hem d'executar aquest codi:

Php artisan migrate

A més a més per a que passport pugui generar access tokens necessita de clau pública i privada per la qual cosa necessitarem generar-les teclejant:

Php artisan passport:install

Ens deuria d'aparèixer una imatge semblant a aquesta:

```
Encryption keys generated successfully.  
Personal access client created successfully.  
Client ID: 1  
Client secret: aYMD5khx0TQc3ynNzrZKXCa77ys9fJ7GMUFCX3TB  
Password grant client created successfully.  
Client ID: 2  
Client secret: dFbFfB7K2KVBZzW3yFzIuYUMJEPMNT4zvsVqerBQ
```

IL-LUSTRACIÓ 30 CLAUS PASSPORT INSTAL·LADES

Ara és el torn de configurar Passport. Cal anar al model App\User i afegir

Laravel\Passport\HasApiTokens

I també hem d'introduir:

Use Notifiable, HasApiTokens;

A la classe User.

El següent pas és habilitar les routes cridant al mètode **Passport::routes** dins del **AuthServiceProvider**.

I per últim hi ha que anar a config/auth.php i assignar el 'driver' => 'passport' per a l'authentication guard anomenada 'api'.

També en l'annex s'explica com s'ha instal·lat XDEBUG que és una ferramenta de debug per a Laravel molt útil per saber què passa amb el codi.

12.1.2. Rutes i comunicacions internes

Com ja hem definit anteriorment el Back-End està constituït per les routes, els models i els controladors. D'aquesta manera les gestions de les peticions al client es fan d'una forma ràpida i ordenada. En aquest apartat definirem algunes rutes que tindrà la api. Aquest fitxer serà l'encarregat en el moment de que arribe una petició del client executar-la o no en funció de la seguretat. En la següent imatge es mostra com s'han definit algunes rutes en el fitxer api.php de l'aplicació.

```
Route::middleware('auth:api')->group(function () {
    Route::post('/teachers/{userId}/subjects/{subjectId}/file/upload', 'API\ResourceController@uploadfile');
    Route::post('/users/register', 'API\AuthController@register');

    Route::post('/warnings/new', 'API\WarningController@create');
    Route::post('/abouts/add', 'API>AboutController@addAboutInformation');
    //USER CONTROLLER
    Route::get('/users/all', 'API\UserController@index');
    Route::get('/users/{id}', 'API\UserController@show');
    Route::get('/users/delete/{id}', 'API\UserController@destroy ');
    //TEACHER CONTROLLER
    Route::get('/teachers/all', 'API\TeacherController@index');
```

```
Route::group(['middleware' => 'auth:api', 'prefix' => 'auth'], function () {

    Route::post('register', 'API\AuthController@register');
    Route::get('user', 'API\AuthController@user');
});
```

IL·LUSTRACIÓ 31 ALGUNES RUTES DEFINIDES EN EL FITXER API.PHP

Com es pot observar, aquestes dos captures del fitxer **api.php** estan englobades per un middleware general, que engloba a totes les consultes on estan. El middleware **auth:api** és el middleware d'autenticació OAuth2 de Passport. Introduint prefix =>auth el que aconseguim és que la ruta tinga abans del registre el prefix /auth/registre o /auth/user.

També es pot saber de forma senzilla des del terminal totes les routes d'accés. Per a saber tots els mètodes d'un determinat Controlador, cal introduir el següent codi:

\$php artisan route:list --name=students i ens mostrarà la següent imatge:

```
PS F:\Developer\Xampp\htdocs\www\tinaferre\backend> php artisan route:list --name=students
Xdebug: [Step Debug] Time-out connecting to debugging client, waited: 200 ms. Tried: localhost:9000 (through xdebug.client_host/xdebug.client_port)
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	api/students	students.index	App\Http\Controllers\API\StudentController@index	api cors auth:api
	POST	api/students	students.store	App\Http\Controllers\API\StudentController@store	api cors auth:api
	GET HEAD	api/students/create	students.create	App\Http\Controllers\API\StudentController@create	api cors auth:api
	GET HEAD	api/students/{student}	students.show	App\Http\Controllers\API\StudentController@show	api cors auth:api
	PUT PATCH	api/students/{student}	students.update	App\Http\Controllers\API\StudentController@update	api cors auth:api
	DELETE	api/students/{student}	students.destroy	App\Http\Controllers\API\StudentController@destroy	api cors auth:api
	GET HEAD	api/students/{student}/edit	students.edit	App\Http\Controllers\API\StudentController@edit	api cors auth:api

IL·LUSTRACIÓ 32 ALCANÇ DE LES RUTES DONAT UN NOM

12.1.3. Middleware

Pel que fa al middleware, s'ha creat un role middleware que el retorna si tens un role específic, és a dir, en el fitxer api.php li afegeixes ['middleware' => 'role:admin'] i si l'usuari té el rol admin, accepta la petició.

El middleware que acabem de definir es detalla en la següent imatge:

```
namespace App\Http\Middleware;

use Closure;

class RoleMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next, $role, $permission = null)
    {
        if (!$request->user()->hasRole($role)) {
            abort(404);
        }
        if ($permission !== null && !$request->user()->can($permission)) {
            abort(404);
        }
        return $next($request);
    }
}
```

IL·LUSTRACIÓ 33 CODI CORRESPONENT AL MIDDLEWARE ROLE

També s'ha creat un altre middleware per a treballar en localhost, aquest middleware anomenat cors, serveix per a permetre l'accés de les peticions http des del client al servidor, ja que al estar treballant des d'un altre servidor, ens denegava l'accés i com tot el desenvolupament s'ha realitzat en localhost s'ha creat un middleware per permetre l'accés. Una vegada estiga en el servidor real hi hauria que llevar-ho.

12.1.4. Models

A l'aplicació de l'escola es troben aquests models: About, Contact, Course, Evaluation, Permission, Resource, Role, Student, Subject, Teacher, User, Warning. Ens centrarem en la dinàmica d'un en concret, ja que els altres fan la mateixa funcionalitat. Explicaré breument en què consisteix el model User.

Per a crear un Model utilitzem el següent codi:

Php artisan make:Model User

Ens apareixerà el model User creat en /app/Http/User.php

En aquest projecte s'ha creat una carpeta específica per a Models, per tindre més organitzats tots els models en /app/Http/Models/User.php

Ha de tindre la ruta per a que Laravel pugui localitzar-lo.

```
namespace App\Models;
```

En aquest model, definirem tots els atributs de la taula de base de dades, definint **protected \$table="**nomb de la taula que correspon el model", **protected \$primaryKey="**definint quin és el nom de la clau primària", **protected \$fillable="**els atributs que vols que es mostren", **protected \$hidden="**els atributs que no vols que es mostren" i per últim **protected \$casts =** "per estructurar com vols que es mostre".

A continuació vejam el codi que té el nostre model.

```
protected $table = 'users';
protected $primaryKey = 'user_id';
protected $fillable = [
    'name', 'surname', 'secondSurname', 'direction', 'telephone', 'email', 'dateofBirth', 'lowDate', 'password', 'dni',
];
protected $hidden = [
    'password', 'remember_token', 'created_at', 'updated_at', 'lowDate'
];
protected $casts = [
    'email_verified_at' => 'datetime',
];
```

Una vegada definits els paràmetres de la taula, ens queda afegir les relacions que té la taula users en altres taules.

En Laravel existeixen diferents tipus de relacions que podem definir. Les relacions presents en el model User són:

```
//1 Usuario tien muchos Rol
public function roles()
{
    //Model|taula-intermitja|fkdelmodel|fkdelataula a que va
    return $this->belongsToMany(Role::class,'role_user','fk_user_id','fk_role_id')-
>withTimestamps();
}

public function permissions()
{
    return $this-
>belongsToMany(Permission::class,'permission_user','fk_user_id','fk_permission_id')-
>withTimestamps();
}

public function students() {
    return $this->hasMany(Student::class)->withTimestamps();
}

public function teachers() {

    return $this->hasMany(Teacher::class)->withTimestamps();
}
}
```

Com podem veure en el codi del Model User, existeix la relació, belongsToMany(Model, Taula, Clau Forànea, Clau Forànea) i la relació hasMany (Model).

- belongsToMany es tracta de relacions Molts a Molts, no és necessari crear una taula intermitja per poder accedir a les dades, definint aquest mètode en el model, aconseguim des del controlador poder accedir a la informació de la taula a la que s'ha definit.
- hasMany es tracta d'una relació Un a Molts, amb esta funció en el model i després en el model que està relacionat ha de tindre en la relació belongsTo, i així ja podríem accedir a les dades des del controlador.

A continuació es mostra l'estructura de les relacions en el Model i en el Model a que apunta (clau foranea).

Relació	Model	Model(clau foranea)
One To Many	\$this->hasMany('Model');	\$this->belongsTo('Model')

Many To Many	<code>\$this->belongsToMany('Model');</code>	<code>\$this->belongsToMany('Model');</code>
---------------------	---	---

Aquestes dos relacions són les que s'han utilitzat al llarg del desenvolupament de l'escola.

12.1.5. Controladors

Cada Model disposa d'un controlador que és l'encarregat de gestionar la informació de la base de dades i mostrar-la cap al client.

Explicarem breument el codi del controlador d'estudiants següent:

```
/**
 * Show subjects of the User from id
 *
 * @param int $id
 * @param \Throwable $exception
 * @return \Illuminate\Http\Response
 */
public function subjects($id)
{
    try {
        Student::findOrFail($id);
        $StudentSubject = Student::with('subjects')->where('student_id', $id)->first();
        return response()->json($StudentSubject->subjects, 200);
    } catch (ModelNotFoundException $exception) {
        return back()->withErrors($exception->getMessage())->withInput();
    }
}
```

IL·LUSTRACIÓ 36 CODI CORRESPONENT AL CONTROLADOR D' ESTUDIANTS

Aquest codi comprova si existeix l' estudiant i fa la consulta a la base de dades per a que li retorne totes les assignatures en les quals està matriculat. Li ho retorna en format JSON. Si no troba l'usuari retorna error.

Un altre exemple del controlador de Student és:

```
public function setSubjectsToStudent(Request $request)
{
    $validator = Validator::make($request->all(), [
        'student_id' => 'required',
        'subjects' => 'required'
    ]);

    if ($validator->fails()) {
        return response()->json(['error' => $validator->errors()->all()], 403);
    }

    $student = Student::where('student_id', $request->student_id)->first();
    if ($student) {
        foreach ($request->subjects as $subj) {
            $subject = Subject::where('subject_id', $subj)->first();
            if (!$subject) {
                return response()->json([
                    'message' => 'La assignatura no existeix'
                ], 403);
            }
            $student->subjects()->attach($subject);
        }
    }
}
```

IL·LUSTRACIÓ 37 CODI CORRESPONENT AL CONTROLADOR D'ESTUDIANTS

Aquest codi, el que fa primerament és validar que l'objecte passat en una crida post des del client siga vàlida. Ho comprova gràcies a l'ús de Validators, el provider d'aquest s'encontra:

```
use Illuminate\Support\Facades\Validator;
```

Si la validació falla, retorna un error en format JSON.

Si no falla la validació, fa la consulta SQL per obtindre les dades de l'estudiant, i si comprova que existeix l'estudiant, aleshores per cada assignatura que es troba en el array de subjects, comprova que existeix l'assignatura en la base de dades, i assigna estudiants a assignatures si no falla l'assignatura. Si no troba l'assignatura retornarà un missatge d'error en format JSON dient que l'assignatura no existeix.

Aquest mètode correspon a l'administrador per poder matricular els estudiants en assignatures enviant com a paràmetres el ID d'estudiant i un Array de IDs de assignatures, per no anar una a una, i fer la gestió més ràpida.

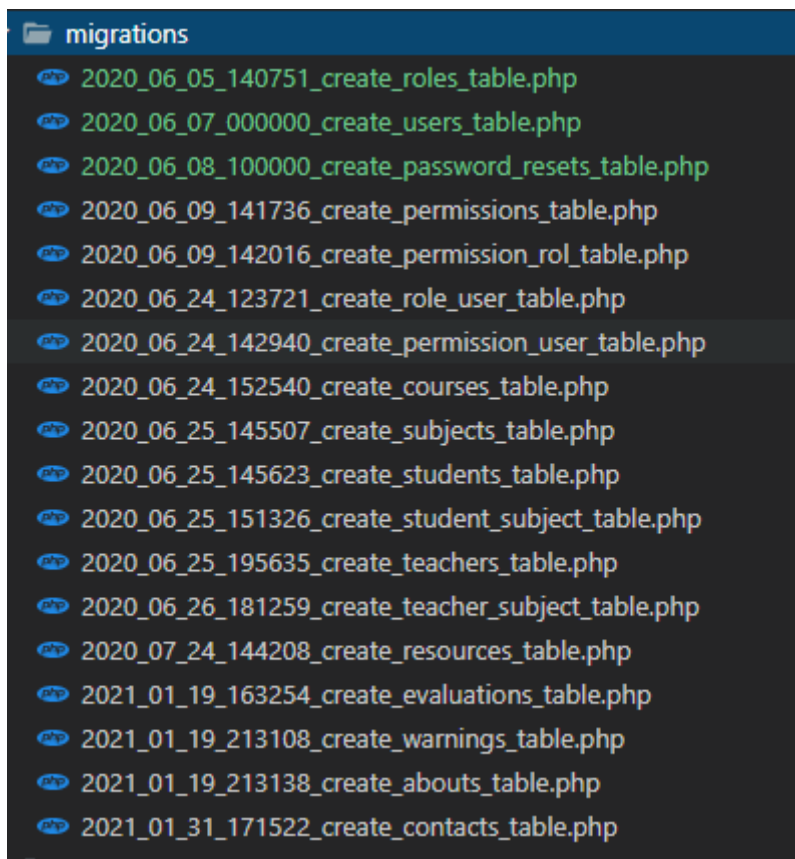
12.1.6. Migracions

Les migracions, són fonamentals per desenvolupar el projecte, Laravel utilitza aquestes migracions per a connectar-les a la base de dades i executar consultes de manera senzilla. Aquestes migracions s'encarreguen de crear les taules de la base de dades.

Per exemple, es crea la migració d'usuari, amb el següent codi:

Php artisan make:migration create_users_table;

I una vegada s'ha creat el fitxer de la migració deu tindre un aspecte com aquest:



IL·LUSTRACIÓ 38 MIGRACIONS DE L'APLICACIÓ

És important no canviar el títol d'aquestes taules ja que Laravel utilitza timestamps que permeten a Laravel determinar l'ordre de les migracions, la fotografia de dalt mostra les migracions fetes per al projecte de l'escola.

A continuació es mostra un exemple de migració de la taula Usuaris.

```
class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->bigIncrements('user_id');
            $table->bigInteger('fk_role_id')->unsigned()->nullable()->default(NULL);
            $table->string('dni')->nullable(false);
            $table->string('name');
            $table->string('surname')->nullable();
            $table->string('secondSurname')->nullable();
            $table->enum('gender',['male','female'])->nullable();
            $table->string('direction')->nullable();
            $table->integer('telephone')->nullable();
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->enum('account_status',['active','pending','rejected'])->default('pending')-
            >nullable();
            $table->rememberToken();
            $table->date('dateofBirth')->nullable();
            $table->date('lowDate')->nullable();
            $table->timestamps();

            $table->foreign('fk_role_id')->references('role_id')->on('roles')->onDelete('cascade')-
            >onUpdate('cascade');
        });
    }
}
```

IL·LUSTRACIÓ 39 CODI CORRESPONENT A LA MIGRACIÓ DE USERS

Una vegada ja tenim tots els camps de la taula, el que hi ha que fer és executar la migració per a que cree en la base de dades la taula, amb el codi:

\$php artisan migrate

12.1.7. Seeders

Per finalitzar amb el Back-End, a vegades ens resulta útil crear dades per omplir les taules buides, i és ahí, on Laravel mitjançant els Seeders, ens proporciona una ferramenta per poder emplenar les taules.

Per a crear un Seeder cal introduir:

Php artisan make:seeder NomSeeder

I una vegada emplenat el Seeder, solament hi ha que fer:

Php artisan db:seed --class=NomSeeder

A continuació mostre el codi del Seeder de l'aplicació escola RoleTableSeeder.php on emplene la taula Role i Permissos.

```
class RoleTableSeeder extends Seeder
{
    /** Run the database seeds.
     * @return void
     */
    public function run()
    {
        $student_permission = Permission::where('type', 'see-tasks')->first();
        $teach_permission = Permission::where('type', 'create-tasks')->first();
        $admin_permission = Permission::where('type', 'edit-users')->first();

        $student_role = new Role();
        $student_role->type = 'student';
        $student_role->description = 'Student from school';
        $student_role->save();
        $student_role->permissions()->attach($student_permission);

        $teacher_role = new Role();
        $teacher_role->type = 'teacher';
        $teacher_role->description = 'Teacher for school';
        $teacher_role->save();
        $teacher_role->permissions()->attach($teach_permission);

        $admin_role = new Role();
        $admin_role->type = 'admin';
        $admin_role->description = 'All access for admin';
        $admin_role->save();
        $admin_role->permissions()->attach($admin_permission);
    }
}
```

IL·LUSTRACIÓ 40 CODI CORRESPONENT AL SEEDER ROLETABLE

12.2. Implementació del Front-End

Tal i com hem explicat en el Back-End, abans d'entrar amb detall, sobre el desenvolupament del Front-End, vaig a esmentar els passos que he seguit per poder instal·lar les ferramentes que s'utilitzaran durant el transcurs del desenvolupament del Front-End.

12.2.1. Instal·lació del Front-End

Per a instal·lar Angular, primer que tot necessitem tindre instal·lat NODE.JS, si no el tenim ens descarreguem des de la pàgina oficial de Node: <https://nodejs.org/es/> l'executable i l'instal·larem a l'ordinador.

Una vegada finalitzat l'instal·lador, entrem al nostre IDE de Desenvolupament, al terminal i actualitzem NPM, que és un gestor de paquets de NODE, per a descarregar-nos les últimes dependències més actuals:

Npm install -g npm@latest

Després hi ha que eliminar la caché de NPM amb:

Npm cache clean --force

Desactivem les auditories de NPM per a evitar errors:

Npm set audit false;

Ara instal·lem l'última versió d'angular CLI:

Npm install -g@angular/cli@latest

Ara que ja tenim instal·lat angular CLI ja podem generar un nou projecte de Angular amb:

Ng new frontend

Durant el procés d'instal·lació ens apareixerà un assistent que ens preguntarà si volem auto generar les rutes, hem de dir que sí. I després ens preguntarà quina opció de CSS elegim. Per aquest projecte s'ha elegit la versió SCSS.

El procés d'instal·lació dura uns minuts fins que es complete. Una vegada finalitzat podem executar-lo escrivint:

Ng serve -o

Aquest codi posa en funcionament el projecte i podem visualitzar-lo en <http://localhost:4200>

Ens apareixerà una imatge com la següent:

Welcome to my-app!



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

IL·LUSTRACIÓ 41 PÀGINA D'INICI DE L'APLICACIÓ ANGULAR ACABADA D'INSTAL·LAR.

El següent pas és configurar Git i Bitbucket, per pujar el control de versions. És pot veure al [ANEX]

12.2.2. Servicis

Els servicis és la forma que té angular per poder comunicar-se, mitjançant components o consumint APIS. Els servicis és una forma de treballar de manera síncrona totes les dades i crides. Els servicis s'encarreguen d'extraure dades i els components de presentar la vista de manera correcta.

Per a generar un servici en angular hi ha que escriure:

Ng generate service NomDelServici

A la app de l'escola hi ha diferents servicis, a continuació es mostra el servici per a autenticar l'usuari mitjançant crida a la app.

```
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { HttpClient } from '@angular/common/http';
import { User } from '../core/models/user.model';
import { environment } from 'src/environments/environment.prod';
import { LoginObject } from '../core/models/login-object.model';
@Injectable({
  providedIn: 'root'
})
export class AuthService {

  constructor(private http: HttpClient) { }

  // User registration
  register(user: User): Observable<any> {
    return this.http.post(environment.endpoint + "/register", user);
  }

  // Login
  login(loginObj: LoginObject): Observable<any> {
    return this.http.post<any>(environment.endpoint + "/auth/login", loginObj );
  }
}
```

IL·LUSTRACIÓ 42 SERVICI ON ES CRIDA LA FUNCIÓ LOGIN

Aquest servici quan és cridat des de la pàgina de login:

```
submitLogin() {
  this.submitted = true;
  //stop here if form is invalid
  if (!this.loginForm.invalid) {
    this.loading = true;

    this.authService.login(new LoginObject(this.loginForm.value)).subscribe(
      result => {
        // console.log(result);
        this.responseHandler(result);
      },
      error => {
        this.error = error.error.message;
        console.log(this.error);
        this.loading = false;
      }, () => {
        this.authState.setAuthState(true);
        this.loginForm.reset();
        this.router.navigate(['/home']);
      }
    );
  } else {
    console.log("Hay errores", this.loginForm.errors);
    return;
  }
}
```

IL·LUSTRACIÓ 43 MÈTODE DE LOGIN EN ANGULAR

Fa la crida a <http://apiescola/api/auth/login>, amb el LoginObject que conté l'adreça electrònica i la contrasenya i si la validació ha anat bé en el servidor, retornarà un

resultat. Aquest resultat és passat directament al mètode `responseHandler()` per a guardar el token rebut del Back-End.

Aquesta funció a la volta crida al servici `token.service` per a treballar amb el token, és a dir, aquest token guarda el token en `localStorage` i conté diferents mètodes per comprovar el estat del token, a l'annex es pot veure el codi d'aquest servici.

12.2.3. Token Interceptor

A l'aplicació de l'escola resulta molt útil per a qualsevol crida http tindre la capçalera de `authorization: bearer + token`, necessària per poder fer crides a la api, ja que sense aquesta capçalera la api no et permet l'accés ja que no estàs autenticat.

Aquesta pàgina typescript s'encarrega d'interceptar totes les crides que es fan des dels servicis a l'API i li afegeix les capçaleres `Authorization`, `Content-Type`, i `Accept`. A continuació es mostra el codi:

```
import { Injectable } from "@angular/core";
import { HttpInterceptor, HttpRequest, HttpHandler, HttpEvent } from "@angular/common/http";
import { TokenService } from "../shared/token.service";
import { Observable } from 'rxjs';

@Injectable()

export class AuthInterceptor implements HttpInterceptor {
  constructor(private tokenService: TokenService) { }

  intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
    const accessToken = this.tokenService.getToken();
    req = req.clone({
      setHeaders: {
        Authorization: "Bearer " + accessToken,
        'Content-Type': 'application/json',
        'Accept' : 'application/json'
      }
    });
    return next.handle(req);
  }
}
```

IL·LUSTRACIÓ 44 CODI TOKEN INTERCEPTOR

Aquest codi intercepta totes les crides http i injecta els headers anteriorment descrits. A continuació es pot veure com des de l'aplicació en funcionament mostra aquestes capçaleres.

12.2.5. Routing Module

És una potent ferramenta donada per angular que s'encarrega de la navegació de l'aplicació. A continuació és mostra el codi del Routing per a l'aplicació web.

```
const routes: Routes = [
  { path: 'home', pathMatch: 'full', component: HomeComponent, canActivate: [AuthGuard] },
  { path: 'setting', component: SettingComponent, canActivate: [AuthGuard] },

  { path: 'admin', component: AdminComponent, canActivate: [AuthGuard], data: { roles: [Role.Admin] }},
  { path: 'subjects', component: SubjectsAdminComponent, canActivate: [AuthGuard], data: { roles: [Role.Admin] }},
  { path: 'subjects-user', component: SubjectsUserComponent, canActivate: [AuthGuard], data: { roles: [Role.Admin] } },

  { path: 'subject/:id', component: SubjectComponent, canActivate: [AuthGuard], },
  { path: 'subject/:id/contact', component: ContactComponent, canActivate: [AuthGuard], },
  { path: 'subject/:id/about', component: AboutComponent, canActivate: [AuthGuard], },
  { path: 'subject/:id/resource', component: ResourceComponent, canActivate: [AuthGuard], },
  { path: 'subject/:id/marks', component: QualificationComponent, canActivate: [AuthGuard], },

  { path: 'login', component: LoginComponent },
  { path: '**', redirectTo: 'home' }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

IL·LUSTRACIÓ 47 CODI MÒDUL DE NAVEGACIÓ

Si no troba ninguna ruta es redirigeix a /home.

12.2.6. Internacionalització

Aquesta aplicació té internacionalització, és a dir, l'usuari pot elegir l'idioma que vol mostrar, per defecte està en valencià. Per a instal·lar aquest mòdul s'han seguit els passos següents:

npm install @ngx-translate/core @ngx-translate/http-loader --save

Una vegada instal·lat, es modifica el app.module.ts per a que quede d'aquesta forma:

```
export function createTranslateLoader(http: HttpClient) {
  return new TranslateHttpLoader(http, './assets/i18n/', '.json');
}
```

I en els imports:

```
TranslateModule.forRoot({
  loader: {
    provide: TranslateLoader,
    useFactory: (createTranslateLoader),
    deps: [HttpClient]
```

```
}  
}},
```

Ara anem a `app.component.ts` i escrivim el següent codi:

```
constructor(translate: TranslateService) {  
  //this language will be used as a fallback when a translation isn't found in the current language  
  translate.setDefaultLang('ca');  
  //the lang to use, if the lang isn't available, it will use the current loader to get them  
  translate.use('ca');  
}
```

Al tractar-se d'una aplicació d'una sola pàgina o (SPA), sols cal introduir aquest codi en l'arrel i els altres components ja hereten l'informació.

Aquest codi per defecte ens diu que el fitxer per a la internacionalització és `ca.json`. En `./assets/i18n/` podem trobar el fitxer `jsons` encarregats de la internacionalització, `ca.json` té el següent aspecte:

```
{  
  "sidebar": {  
    "resources": "Recursos",  
    "marks": "Qualificacions",  
    "contact": "Contacte",  
    "about": "Sobre"  
  }...{..}
```

Aquest JSON de clau valor és imprescindible per a que després des del component `html` amb la crida següent es mostre el valor.

Exemple de Crida de Internacionalització: `{{ sidebar.resources | translate }}`

Actualment hi ha dos idiomes disponibles el valencià i el castellà.

12.2.7. Seguretat

A banda de la seguretat de l'API i l'ús amb token en totes les crides HTTP que s'ha vist en punts anteriors, s'ha fet una guia per a implementar el protocol HTTPS a la part del Front-End. Aquest protocol ens mostra que una connexió és segura i ens ofereix tres capes de protecció. Aquestes capes son: Xifrat de dades, Integritat de dades i autenticació.

Ja que al treballar amb `localhost` no podem crear un certificat fiable, s'ha creat un certificat explícit per a treballar amb `localhost`, és a dir, està signat per mi mateixa en lloc d'estar signat per Trusted Certificate Authority (CA). Per defecte aquest certificat no és



Aplicació multiplataforma per a la gestió d'una escola musical

confiable, nosaltres haurem d'afegir el nou certificat generat per Trusted Root Certification Authorities després, quan s'implemente en el entorn de servidor real.

Els passos que he seguit per per aquest certificat són:

-S'ha creat el fitxer certificate.cnf. A continuació es pot veure el que conté:

```
[req]
default_bits = 2048
prompt = no
default_md = sha256
x509_extensions = v3_req
distinguished_name = dn
[dn]
C = ES
ST = Alicante
L = Banyeres de Mariola
O = Escola Tina Ferre
OU = WEB
emailAddress = admin@escolaTF.com
CN = localhost
[v3_req]
subjectAltName = @alt_names[alt_names]
DNS.1 = localhost
```

IL·LUSTRACIÓ 48 EXEMPLE DE CERTIFICAT

-Amb Git bash, generem els certificats amb el següent codi:

```
juan-@TorreMSI MINGW64 /f/Developer/certificates
$ openssl req -new -x509 -newkey rsa:2048 -sha256 -nodes -keyout localhost.key -
days 3560 -out localhost.crt -config certificate.cnf
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'localhost.key'
-----
```

IL·LUSTRACIÓ 49 GENERAR CERTIFICATS AMB OPENSSL

Aquest codi ens haurà generat dos fitxers: localhost.key i localhost.crt.

-Una vegada tenim els certificats cal configurar Angular CLI per a que utilitze aquests certificats. Per a fer-ho hem d'escriure al fitxer Package.json del nostre projecte el següent codi:

```
"start": "ng serve --ssl --ssl-key F:\\Developer\\certificates\\localhost.key --ssl-
cert d:\\certificates\\localhost.crt",
```

-Una vegada editat el script start del package.json per a que utilitze el certificat el següent que hem de fer és autoritzar en el Trusted Root Certification Authorities guardat en Windos, per a que no done errors.

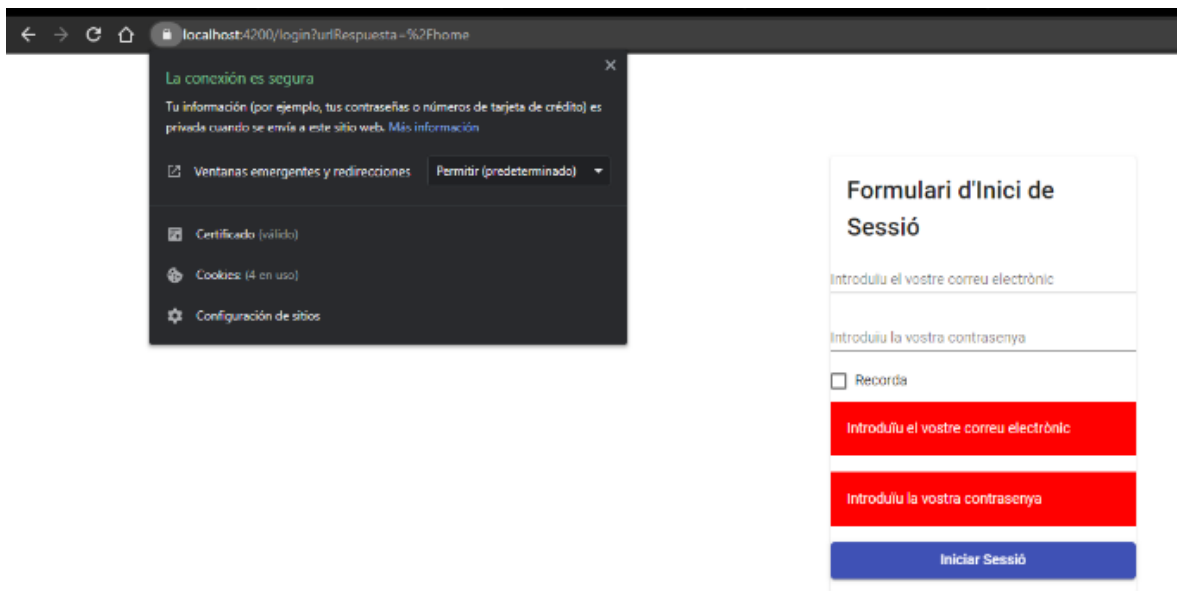
-Obrim el programa Microsoft Management Console i teclegem mmc, seleccionem Certificats i usuari actual i ens mostrarà tots els certificats. Una vega ahí, el que hem de fer és afegir el nostre certificat en "entitats de certificació d'arrel de confiança", i ja estaria.

És una solució temporal fins que generem el nou certificat del servidor real.

Per comprovar que tot ha anat bé anem al projecte i teclegem en el terminal:

Npm start

Ens obrirà l'aplicació amb el protocol HTTPS ja, tal i com podem observar en aquesta imatge:



IL·LUSTRACIÓ 50 PÀGINA WEB D'INICI ON ES VEU L'APLICACIÓ SEGURA AMB EL PROTOCOL HTTPS.

12.3. Codi Font

El codi de l'aplicació tant el Front-End com el Back-End es pot trobar al repositori:

<https://juacasa6@bitbucket.org/juacasa6/backend.git>

<https://juacasa6@bitbucket.org/juacasa6/frontend.git>

13. Pressupost

A continuació es detallarà el pressupost necessari per a la realització del projecte. Aquest pressupost està dividit en tres parts. El Maquinari o cost de material, utilitzat per poder realitzar el projecte, el programari i el cost personal per pagar un jornal a un informàtic. També hi haurà un pressupost final englobant aquests tres costos.

13.1. Maquinari o costs materials

Unitats	Concepte		Preu unitari	Subtotal
1	Placa Base	MSI Z270 Gaming M5 (M5-7A78)	137,64€	137,64€
4	RAM	Corsair 1x8Gb 3200MHz DDR4	48,94€	195,76€
1	CPU	Intel i7-7700K CPU 4.20Ghz	271€	271€
1	SSD	860 EVO M.2 500Gb	75,41€	75,41€
1	HDD	Toshiba 2Gb	43,58€	43,58€
1	Pantalles auxiliars	Dell 27"	189€	189€
		Acer 24"	129€	129€
1	Ordenador Portàtil	Msi G62 Apache Pro	800€	800€
			Total amb IVA(21%)	1841,39€

13.2. Programari o costos tècnics

En aquest apartat es presentaran es costos tècnics del programari que s'han utilitzat per posar en marxa l'aplicació.

Unitats	Concepte	Preu Unitari	Subtotal
1	Sistema Operatiu Windows 10 Home 64bits	109,29€	109,29€
1	Microsoft Office 2020, Llicència estudiant	0,0€	0,0€
1	Visual Studio Code	0,0€	0,00€
		Total amb IVA(21%)	109,29€

13.3. Cost Personal

Per al cost Personal s'ha calculat a partir de la data d'inici del projecte i la data final del projecte. A continuació vorem les diferents fases d'aquest projecte:

Anàlisi de Projecte	01/07/2020	14/08/2020
Requisits, funcionals (Objectius)	01/07/2020	28/07/2020
Recerca d'Informació (Estat de l'art)	02/07/2020	30/07/2020
Proves API REST (Arquitectura de l'aplicació)	22/07/2020	11/08/2020
Inici Documentació	05/08/2020	10/08/2020
Planificació del Projecte	10/08/2020	14/08/2020
Disseny del Projecte(Estructura de l'API)	17/08/2020	05/10/2020
Disseny de pantalles	17/08/2020	04/09/2020
Fer prototips WEB	01/09/2020	07/09/2020
Fer Base de Dades	07/09/2020	25/09/2020
Instal·lar Laravel i Angular	28/09/2020	02/10/2020
Creació d'index	05/10/2020	05/10/2020
Implementació del Projecte	06/10/2020	21/12/2020
Dessenvolupament Back-END	06/10/2020	21/12/2020
Dessenvolupament Front-END	06/10/2020	21/12/2020
Css Frontend	18/11/2020	30/11/2020
Revisió d'errors	01/12/2020	07/12/2020
Fin d'implementació	07/12/2020	07/12/2020
Proves Projecte	10/12/2020	14/12/2020
prova web	08/12/2020	28/12/2020
prova disseny responsive	10/12/2020	11/12/2020
fin proves	14/12/2020	14/12/2020
escriure memòria	15/12/2020	11/01/2021
fin memòria	12/01/2021	12/01/2021



Aplicació multiplataforma per a la gestió d'una escola musical

S'ha calculat tenint en compte que els costos s'extrauran del cost d'execució sobre les hores dedicades a la realització.

En total s'han dedicat 6 mesos a aquest projecte, un total de 132 dies laborals descomptant caps de setmana i festius. Tenint en compte que la jornada laboral s'estima en 6 hores cada dia.

Hores treballades = dies x jornades = 132 dies x 6h cada dia = 792h

I que el sou per hora d'un enginyer Informàtic a Espanya és 13,58€/hora.

Unitats(hores)	Concepte	Preu Unitari	Subtotal
1	Honoraris de I.T.	13,58€	10.755,36€
		Total amb Iva(21%)	13.013,98€

13.4. Pressupost Final

Una vegada tenim tots els pressupostos, es detallarà a continuació la proposta final de pressupost.

Concepte	Pressupost
Costs Materials (Maquinari)	1841,39€
Costs Tècnics (Programari)	109,29€
Honoraris	13.013,98€
Total amb Iva(21%)	14.964,66€

El projecte "d'aplicació web multiplataforma per a la gestió d'una escola musical" ascendeix a la quantitat de CATORZE MIL NOU-CENTS SEIXANTA-QUATRE euros I SEIXANTA-SIS Cèntims.

14. CONCLUSIONS

14.1. Conclusions.

Una vegada finalitzat el projecte és moment de reflexionar sobre el que hem obtingut al realitzar aquest projecte.

En aquest TFG s'ha realitzat una aplicació web i una api, des de zero per donar suport a una escola musical que no tenia cap tipus de plataforma.

Després dels requeriments que el centre exigia, es va profunditzar en prendre la millor solució possible. Es va realitzar un treball metòdic a nivell de pressupost, planificació, metodologia, objectius, arquitectura, es van realitzar diferents mockups per veure el disseny de l'aplicació, crear la composició de la base de dades, tecnologies per al desenvolupament entre altres.

Des d'un principi aquestes tecnologies usades eren noves, la corba d'aprenentatge va costar un poc, en algunes ocasions, però al tractar-se de Frameworks tan populars, hi ha tot una comunitat donant-te suport i gran quantitat de recursos al teu abast.

En quant a la funcionalitat principal de la web aconseguida, és la d'oferir una versió 1.0, a la comunitat educativa permetent als alumnes i als professors, pujar el treball que realitzen i establir comunicacions interpersonals entre professor i alumne, i mantindre al alumne incentivat, per incrementar el valor de l'aprenentatge.

Respecte als resultats, els objectius s'han aconseguit, i durant el procés, he après dos tecnologies més. Ara sols queda implementar-ho i recollir el feed-back. I al tractar-se d'una versió 1.0 recollir errors i anant solucionant-los i anar millorant l'aplicació.

Per a finalitzar m'agradaria anyadir que aquesta aplicació serà molt útil davant la complicada situació sanitària que estem fent front. Els usuaris de l'aplicació, tant alumnes com professors, tenen al seu abast una ferramenta útil per poder continuar formant-se i gaudint de la música, i així continuar avançant en el seu aprenentatge.

14.2. Treball futur.

En aquest punt, s'explicaran algunes propostes que podrien ser interessants de cara al futur:

- Creació d'un fòrum on tots els alumnes d'una assignatura concreta puguin comunicar-se, ja que ara sols poden comunicar-se amb el professor.
- Convertir la app angular en una aplicació per a mòbil utilitzant el framework iònic, ja que angular està preparat per poder migrar a una aplicació mòbil.
- Introduir un nou rol Music, per gestionar actes que vols fer com a músic quan estàs dins de la banda.
- Afegir l'opció de poder fer exàmens en línia.



15. BIBLIOGRAFIA

- [1] "Laravel Docs" <https://laravel.com/docs/7.x>
- [2] "Laravel 7 i Passport" <https://programacionymas.com/blog/api-rest-laravel-passport>
- [3] "Angular Docs" <https://angular.io/docs>
- [4] "Que es moodle y para que sirve?" <https://www.maximaformacion.es/e-learn/que-es-moodle-y-para-que-sirve/>
- [5] "moodle" <https://moodle.org/?lang=es>
- [6] "Que es y como dispones de xDebug" <https://desarrolloweb.com/articulos/que-es-instalar-configurar-xdebug.html>
- [7] "stackoverflow" <https://es.stackoverflow.com/>
- [8] "tractament de dades" https://www.boe.es/boe_catalan/dias/2018/12/06/pdfs/BOE-A-2018-16673-C.pdf
- [9] "arquitectura api rest" <https://juanda.gitbooks.io/webapps/content/api/arquitectura-api-rest.html>
- [10] "que es google classroom" <https://www.mundocuentas.com/google/classroom/>
- [11] "que es edmodo" <https://bit4learn.com/es/lms/edmodo/>
- [12] "laravel" <https://desarrolloweb.com/home/laravel>
- [13] "laravel" <https://styde.net/>
- [14] "xDebug" <https://es.wikipedia.org/wiki/Xdebug>
- [15] "hacer debug en laravel con xdebug y phpstorm" <https://www.cursosdesarrolloweb.es/blog/debug-en-laravel-con-xdebug-y-phpstorm/>

16. Índex d'Il·lustracions

IL·LUSTRACIÓ 1 - ARQUITECTURA	19
IL·LUSTRACIÓ 2 Diagrama de Gantt	23
IL·LUSTRACIÓ 3. Patró de disseny MVC de LARAVEL	24
IL·LUSTRACIÓ 4 Exemple d'autenticació basada en token	26
Il·lustració 5 Model de la base de dades de l'escola	28
Il·lustració 6 Estructura de carpetes del back-end	30
IL·LUSTRACIÓ 7 – Estructura de carpetes del Front-end.....	32
IL·LUSTRACIÓ 8 Versió escriptori (Rol Estudiant).....	33
IL·LUSTRACIÓ 9 Pàgina de login de l'aplicació web.....	34
IL·LUSTRACIÓ 10 Pàgina d'inici de l'aplicació web	34
IL·LUSTRACIÓ 11 La pàgina de l'assignatura1->Recursos.....	35
IL·LUSTRACIÓ 12 La pàgina d' inici de l'assignatura	35
IL·LUSTRACIÓ 13 Qualificacions de l'assignatura 1	36
IL·LUSTRACIÓ 14 Pàgina de contacte de l'assignatura 1	36
IL·LUSTRACIÓ 15 Pàgina de About de l'assignatura 1	37
IL·LUSTRACIÓ 16 Pàgina de Perfil de l'usuari	37
IL·LUSTRACIÓ 17 Flux d'un professor pujant un recurs	38
IL·LUSTRACIÓ 18 Flux d'un professor pujant una nota	39
IL·LUSTRACIÓ 19 Flux on un professor respon a un estudiant.....	40
IL·LUSTRACIÓ 20 Flux on un professor puja un nou about	41
IL·LUSTRACIÓ 21 Flux d'administrador afegint un nou usuari	42
IL·LUSTRACIÓ 22 Imatge d'assignatures on es mostren (totes o per cursos), segons polsem el botó toggle, i per últim mockup d'afegir assignatura	43
IL·LUSTRACIÓ 23 Relació d'usuaris-assignatures on, a cada usuari, se li assigna una assignatura	44
Il·lustració 24 Disseny d'aplicació web per a tablet	45
IL·LUSTRACIÓ 25 Flux de navegació d'un usuari de tipus Estudiant. En un smartphone-> Max-width: 480px.....	46
Il·lustració 26 Diagrama de funcionament del rol estudiant	47
Il·lustració 27 Diagrama de funcionament del rol professor	47
Il·lustració 28 Diagrama de funcionament del rol administratiu	48
IL·LUSTRACIÓ 29- Pàgina d'inici del projecte creat en laravel	50
IL·LUSTRACIÓ 30 Claus Passport instal·lades	51
Il·lustració 31 Algunes routes definides en el fitxer api.php.....	52
Il·lustració 32 Alcanç de les routes donat un nom.....	53
Il·lustració 33 Codi corresponent al middleware Role	53
Il·lustració 34 Codi corresponent al Model User.....	55
Il·lustració 35 Codi corresponent al model user	55
Il·lustració 36 Codi corresponent al controlador d' estudiants.....	56
Il·lustració 37 Codi corresponent al controlador d'estudiants	57
Il·lustració 38 Migracions de l'aplicació.....	58
Il·lustració 39 Codi corresponent a la migracio de users	59



Il·lustració 40 Codi corresponent al seeder Roletable	60
IL·LUSTRACIÓ 41 Pàgina d'inici de l'aplicació angular acabada d'instal·lar.	62
Il·lustració 42 Servici on es crida la funció login.....	63
Il·lustració 43 Mètode de login en angular	63
Il·lustració 44 Codi Token Interceptor.....	64
Il·lustració 45 Capçalera de l'aplicació web on es veu el token	65
Il·lustració 46 Codi authguard	65
Il·lustració 47 Codi mòdul de navegació	66
IL·LUSTRACIÓ 48 Exemple de certificat.....	68
IL·LUSTRACIÓ 49 Generar certificats amb openssl.....	68
IL·LUSTRACIÓ 50 Pàgina web d'inici on es veu l'aplicació segura amb el protocol https.	69
IL·LUSTRACIÓ 51 ANNEX TOKEN SERVICE codi.....	77
IL·LUSTRACIÓ 52 Requadre on es mostra tota l'informació de PHP	78
IL·LUSTRACIÓ 53 Informació després d'introduir phpInfo en el wizard	79
IL·LUSTRACIÓ 54 Configuració xdebug en el Ide	80

17. ANNEXOS

17.1. Annex token service

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class TokenService {

  private issuer = {
    login: 'http://apiescola/api/auth/login',
    register: 'http://apiescola/api/auth/register'
  }

  constructor() { }

  handleData(token){
    localStorage.setItem('auth_token', token);
  }

  getToken(){
    return localStorage.getItem('auth_token');
  }

  // Verify the token
  isValidToken(){
    const token = this.getToken();

    if(token){
      const payload = this.payload(token);
      if(payload){
        return Object.values(this.issuer).indexOf(payload.iss) > -1 ? true : false;
      }
    } else {
      return false;
    }
  }

  payload(token) {
    const jwtPayload = token.split('.')[1];
    return JSON.parse(atob(jwtPayload));
  }

  // User state based on valid token
  isLoggedIn() {
    return this.isValidToken();
  }

  // Remove token
  removeToken(){
    localStorage.removeItem('auth_token');
  }
}
```


IL·LUSTRACIÓ 51 ANNEX TOKEN SERVICE CODI

17.2. Annex instal·lació Xdebug

Xdebug és una extensió de PHP que ens proporciona la capacitat de depuració de codi i errors. S'ha usat per depurar la api de l'escola, per a instal·lar-la al projecte s'han seguit els següents passos:

Primer que res, hem de saber que versió utilitzem de php, per saber-ho hem d'engegar el servidor apache. Una vegada el servidor esta arrancat, hem d'anar al navegador i introduir localhost/dashboard el qual ens dirigira a la pàgina de Xamp.

Una vegada estem ahí, hem de polsar el botó PHPinfo i ens apareixerà una informació com la següent:

PHP Version 7.4.11 	
System	Windows NT TORREMSI 10.0 build 19042 (Windows 10) AMD64
Build Date	Sep 29 2020 13:11:07
Compiler	Visual C++ 2017
Architecture	x64
Configure Command	cmd /c "php --enable-snapshot-build --enable-debug-pack --with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1sdk,shared --with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1sdk,shared --enable-object-out-dir=.\obj --enable-com-dotnet=shared --without-analyzer --with-pgsql"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	F:\Developer\Xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,TS,VC15
PHP Extension Build	API20190902,TS,VC15
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

IL·LUSTRACIÓ 52 REQUADRE ON ES MOSTRA TOTA L'INFORMACIÓ DE PHP

Acte seguit, hem d'entrar en mode desenvolupador i copiar tot el contingut de la pàgina per a pegar després les dades en el requadre de text de la pàgina <https://xdebug.org/wizard>, que si tot ha anat bé, ens deuria de mostrar els passos a seguir per completar la instal·lació. Ens mostrara el següent:

Installation Wizard

Summary

- **Xdebug installed:** 3.0.2
- **Server API:** Apache 2.0 Handler
- **Windows:** yes
- **Compiler:** MS VC15
- **Architecture:** x64
- **Zend Server:** no
- **PHP Version:** 7.4.11
- **Zend API nr:** 320190902
- **PHP API nr:** 20190902
- **Debug Build:** no
- **Thread Safe Build:** yes
- **OPcache Loaded:** no
- **Configuration File Path:** no value
- **Configuration File:** F:\Developer\Xampp\php\php.ini
- **Extensions directory:** F:\Developer\Xampp\php\ext

You're already running the latest Xdebug version

But here are the instructions anyway:

1. Download [php_xdebug-3.0.2-7.4-vc15-x86_64.dll](#)
2. Move the downloaded file to F:\Developer\Xampp\php\ext
3. Update F:\Developer\Xampp\php\php.ini and change the line
`zend_extension = F:\Developer\Xampp\php\ext\php_xdebug-3.0.2-7.4-vc15-x86_64.dll`
4. Restart the webserver

IL·LUSTRACIÓ 53 INFORMACIÓ DESPRÉS D'INTRODUIR PHPINFO EN EL WIZARD

Apareix com a que ja està instal·lat, però les instruccions per reinstal·lar-lo també estan, sols tindriem que seguir les instruccions per a instal·lar-.

Per a configurar el nostre IDE, per a que treballi amb Xdebug, hem d'editar el fitxer launch.json d'aquesta forma:

```
"version": "0.2.0",
"configurations": [
  {
    "name": "Listen for XDebug",
    "type": "php",
    "request": "launch",
    "port": 9000
  },
  {
    "name": "Launch currently open script",
    "type": "php",
    "request": "launch",
    "program": "${file}",
    "cwd": "${fileDirname}",
    "port": 9000
  }
]
```

IL·LUSTRACIÓ 54 CONFIGURACIÓ XDEBUG EN EL IDE

Una vegada fet, ja quedaria operatiu el debugger de PHP.