

---

This is the **published version** of the bachelor thesis:

Chóez Bolaños, Stefany Ariana; Tallo Sendra, Marc, dir. Covid Control Agenda.  
2021. (958 Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/238436>

under the terms of the  license

# Covid Control Agenda

Stefany Ariana Chóez Bolaños

**Resum–** Covid Control Agenda és una aplicació dirigida al sector de l'educació per a ajudar a les escoles, instituts i universitats en aquests temps de pandèmia en el qual estem vivint, gràcies a Covid Control Agenda podrem identificar i actuar de manera ràpida davant d'un cas de Covid-19 mitjançant la creació de casos i avisos els quals rebran pares i alumnes. En aquest informe podrem observar el desenvolupament de l'aplicació que serà una aplicació de tipus PWA(Progressive Application Web).

**Paraules clau–** anàlisi, covid, disseny, implementació, pwa

**Resumen–** Covid Control Agenda es una aplicación dirigida al sector de la educación para ayudar en las escuelas, institutos y universidades en estos tiempos de pandemia en el cual estamos viviendo, gracias a Covid Control Agenda podremos identificar y actuar de manera rápida ante un caso de Covid-19 mediante la creación de casos y avisos los cuales recibirán padres y alumnos. En este informe podremos observar el desarrollo de la aplicación que será una aplicación de tipo PWA(Progressive Application Web).

**Palabras claves–** análisis, covid, diseño, implementación, pwa

**Abstract–** Covid Control Agenda is an application aimed at the education sector to help schools, institutes and universities in these times of pandemic in which we are living, thanks to Covid Control Agenda we can identify and act quickly in the face of a case of Covid-19 through the creation of cases and notices which parents and students will receive. In this report we will be able to see the development of the application which will be a PWA (Progressive Application Web) application.

**Keywords–** analysis, covid, design, implementation, pwa



## 1 INTRODUCCIÓ

**C**OVID Control Agenda és una aplicació pensada per a ajudar a les escoles, col·legis, instituts i fins i tot a les universitats a adaptar-se, ja que amb la pandèmia que estem vivim l'educació s'ha vist obligada a canviar la manera en la qual s'impartien les classes. I per aquesta situació, ha sortit la idea d'aquesta aplicació Covid Control Agenda, que ens permetrà tenir un control de si algun alumne o algun familiar proper d'algun alumne ha donat positiu en Covid-19, com també poder rebre avisos de canvis d'aula, d'entrada i sortida per tal de garantir la salut de les persones. Aquesta aplicació no serà com la típica aplicació que ens hem de descarregar sinó que serà de tipus PWA(Progressive Application Web) que

són bàsicament pàgines web, però mitjançant l'ús de Service Workers i altres tecnologies es comporten més com a aplicacions natives que com a aplicacions web. En aquest informe s'explicarà la fase final tot el desenvolupament del projecte.

## 2 ESTAT DE L'ART

El projecte es troba en la seva fase final i de lliurament, on ja s'ha acabat d'implementar totes les funcionalitats amb màxima prioritat. Actualment en el mercat no trobem cap aplicació o pàgina web d'aquest tipus, com a molt ens podem trobar aplicacions que tracten sobre la detecció de la Covid-19 o aplicacions que ens ajuden a comunicar-nos amb el professorat i fer una millor planificació dels cursos educatius, és per això que es va plantejar de porta a terme el desenvolupament de l'aplicació Covid Control Agenda, ja que avui en dia amb tot el que hem viscut i s'està vivint a causa de la pandèmia, les escoles/instituts/universitats o fins i tot el sector sanitari els agradaria tenir aquest control dels casos positius de la Covid-19 en els centres educatius. Del

- E-mail de contacte: stefanyariana.choez@e-campus.uab.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Marc Talló Sendra
- Curs 2020/21

projecte destacar que per la part del backend les tecnologies utilitzades han estat:

- Nodejs: per crear el nostre servidor i totes les funcions CRUD que formaran part del backend
- MongoDB: la base de dades està orientada a documents com es va explicar al primer informe.
- Postman: s'ha utilitzat aquesta eina per realitzar peticions i comprovar que les funcions CRUD implementades realitzaven la seva funció.
- Visual Code: eina utilitzada per realitzar la implementació.
- Git Bash: eina utilitzada per executar el servidor.

Per altra banda, la part del front-end s'ha realitzat la implementació amb angular que és un framework per realitzar aplicacions web desenvolupades en TypeScript, gràcies amb angular s'ha pogut posar a prova tot el que es va ensenyar del Model vista controlador a l'assignatura de Tecnologies i desenvolupament web, ja que amb angular podem fer que el desenvolupament i les proves siguin una mica més fàcils gràcies a com està estructurat tot.

### 3 OBJECTIUS

Aquesta aplicació té tres grups objectius a aconseguir:

- Pel professorat: ha de poder informar els pares i als alumnes si s'ha detectat algun cas positiu de Covid-19, així com poder informar de les diferents mesures que el centre ofereix per tal de garantir la salut dels estudiants.
- Pels estudiants: han de poder tenir accés a tota la informació, amb relació les mesures preventives preses en el seu centre per tal de poder complir amb totes elles, i també ha de ser informat si s'ha detectat algun cas de Covid-19
- Pels familiars dels estudiants: poden disposar d'informació del seu centre d'educació amb relació a les mesures contra el Covid-19, així com per exemple, si algun pare vol recollir al seu fill, doncs que pugui saber per quina porta sortirà per tal de garantir certes mesures preventives a causa de la situació actual del Covid-19. També, poder informar-se de si n'hi ha hagut algun cas positiu de la Covid-19 al centre o a un grup de classe. Per altra banda, si el seu fill és menor d'edat, podrà informar-se de l'itinerari que segueix, ja que en moltes escoles i col·legis per tal d'evitar aglomeracions de gent segueixen certes mesures de seguretat com per exemple, a quina hora surten al pati el grup del seu fill o a quina hora sortirà aquell dia.

Així doncs, en general pels usuaris que interactuaran serà una eina que té com a objectius donar aquests resultats.

### 3.1 Descripció del projecte

L'eina a desenvolupar serà una aplicació web orientada a l'ajuda i control de si algun alumne o algun familiar proper d'algun alumne ha donat positiu en Covid-19, el producte haurà de ser accessible des de dispositius mòbils de mitjà-gran format, tauletes i ordinadors.

En primer lloc, s'ha de realitzar un control d'usuari perquè els usuaris com a primer mòdul de funcionalitats puguin accedir i segons els tipus d'usuaris (3 usuaris) puguin tant crear o visualitzar casos sobre si s'ha trobat alguna incidència de positiu en Covid-19, cada cas s'ha de poder especificar a quin curs, classe i grup i alumnes estan implicats o els afecta, per tant, també s'ha de poder crear tota aquesta jerarquia, els professors han de poder crear usuaris tant d'alumnes com de pares/tutors si calgués

Per altra banda, com a segon mòdul també s'haurà de poder crear i visualitzar avisos per si cal gestionar aglomeracions o per informar de les mesures preventives.

Per altra banda, com a tercer mòdul els usuaris (pares i alumnat) han de poder tenir un apartat on puguin comentar els seus dubtes i consultes.

I com a quart mòdul que seria un opcional, els pares podrien veure l'itinerari dels seus fills per tal de mantenir un control de com es realitzen les mesures preventives que s'han de mantenir en aquest temps de pandèmia. Aquest mòdul si per algun inconvenient no s'arribi a dur a terme, quedarà com a futura ampliació.

Això es pot observar clarament amb el diagrama de casos d'ús que es troba en la figura [1].

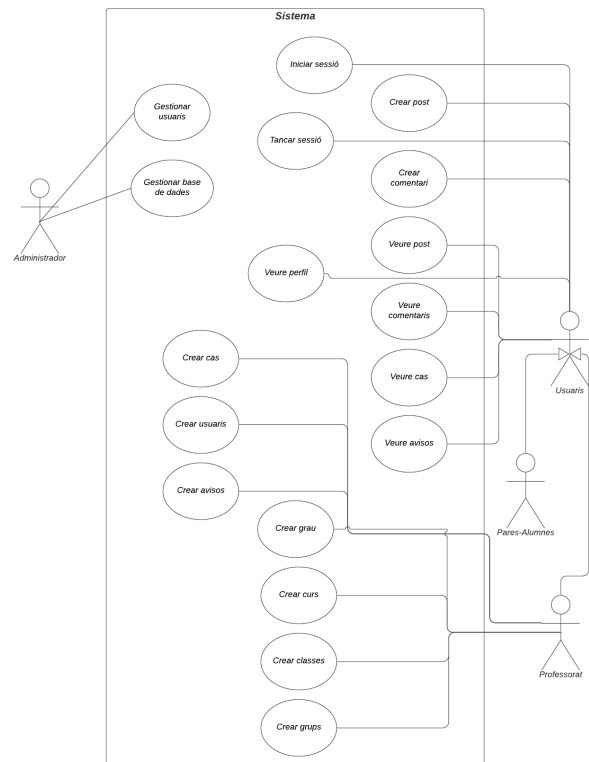


Fig. 1: Diagrama de cas d'ús (Ampliació a l'annex)

### 3.2 Quadre de criticitats

Per altra banda, s'ha assignat a cada funcionalitat que ha de tenir l'eina un nivell de criticitat per tal de gestionar en primer lloc tot allò que tingui una prioritat alta.

Nom del requeriment	Prioritat
Control d'usuaris	1
Crear Cas	1
Posts i comentaris	3
Avisos	2
Crear Curs actual, graus, classea i grups	1
Crear Usuaris	1
Itinerari escolar	4

## 4 METODOLOGIA

Per poder desenvolupar tot el seguit de tasques plantejades del projecte s'ha utilitzat la metodologia àgil Kanban, s'ha decidit fer servir kanban, ja que ens serveix per a visualitzar el treball, evitar l'acumulació de treball pendent i maximitzar l'eficiència. És un procés que permet millorar constantment el flux i la qualitat del treball. A més a més, que Kanban a diferència d'altres metodologies es pot fer servir per projectes en el que només hi participa una persona, ja que la majoria de les metodologies que existeixen estan més pensades al treball en equip i kanban té aquesta flexibilitat que es pugui aplicar tant a equips d'una persona com de més. Per altra banda, s'ha utilitzat trello per portar un control de les tasques que s'han de realitzar.

## 5 PLANIFICACIÓ

Les tasques s'han dividit en sis blocs principals: un orientat a la realització de l'informe inicial i a la formació i anàlisi del projecte, un segon en la definició i planificació del projecte, un tercer centrat en el desenvolupament complet del producte, una quarta per les funcionalitats avançades i la realització de test, una cinquena per funcionalitats extres, i per últim un darrer bloc basat en la redacció exhaustiva del procediment i l'extracció de conclusions de cara el document final. Donat que les tecnologies escollides per realitzar el desenvolupament són relativament noves i no es té cap experiència prèvia la primera fase del projecte és molt important i crítica. La cerca d'informació sobre el framework i les llibreries utilitzades és clau per tal de fer-se una idea sobre quins són els possibles conflictes ens trobarem de cara a la fase de desenvolupament.

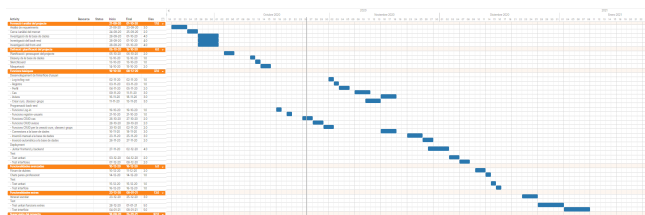


Fig. 2: Diagrama de Gantt(Ampliació a l'annex)

## 6 DESENVOLUPAMENT

A continuació, s'explicarà tot el desenvolupament que s'ha realitzat d'aquest projecte per tal de desenvolupar l'eina.

### 6.1 Rest API

El primer pas realitzat ha estat la creació del back-end, la base de dades que s'ha realitzat és de tipus no relacional amb MongoDB, tant els esquemes com les funcions CRUD s'han desenvolupat amb les tecnologies NodeJS i ExpressJS. L'estructura ha sigut la següent:

- **Models:** En aquesta part s'han creat els models usant Mongoose per a poder guardar la informació en la base de dades seguint l'esquema. Com a base de dades utilitzarem MongoDB. MongoDB és una base de dades Open Source NoSQL orientada a documents tipus JSO.
- **Controllers:** En aquesta part s'ha definit i implementat les funcions CRUD usant Mongoose i els models creats. Les funcions CRUD són:
  - \* Create → Creació de dades
  - \* Read → Lectura de dades
  - \* Update → Actualització de dades
  - \* Delete → Eliminació de dades
- **Routes:** En aquesta part es defineixen les rutes de l'API, es crea un fitxer amb les rutes de cada model creat. Tots els objectes es manipulen mitjançant URI, per exemple, si tenim un recurs usuari i volem accedir a un usuari en concret nostre URI seria /usuari/\*identificadorobjecte, amb això ja tindríem un servei USER preparat per a obtenir la informació d'un usuari, donat un ID.
- **Fitxer server.js:** Aquest serveix per ajuntar les parts comentades anteriorment, aquest fitxer conté:
  - \* Connectar-se a la base de dades de MongoDB
  - \* Carregar el model de llibres
  - \* Agregar el bodyparser.
  - \* Registrar les rutes creades en el server

L'estructura de la base de dades que s'ha utilitzat és la següent:

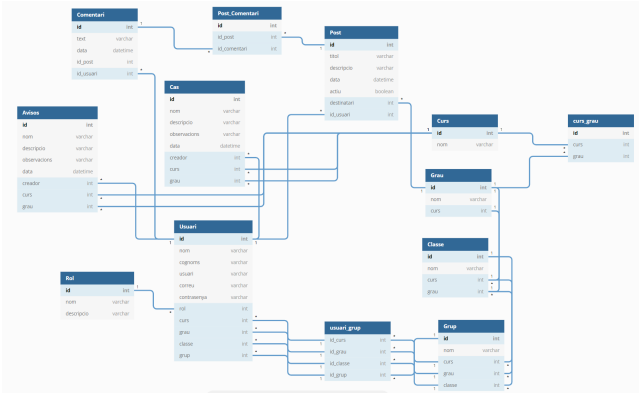


Fig. 3: Disseny base de dades(Ampliació a l'annex)

## 6.2 Creació de l'aplicació web

En aquest apartat s'ha dividit en mòduls segons les diverses funcionalitats que té l'aplicació web.

### 6.2.A Mòdul control d'usuaris

És un mòdul típic per poder accedir com registrar-se a la web omplint formularis, un dels aspectes a destacar d'aquest mòdul és el tractament de la contrasenya com a dada privada. És per això, que s'ha utilitzat la llibreria bcryptjs que a diferència d'altres, porta incorporat un valor anomenat salt, que és un fragment aleatori que s'usarà per a generar el hash associat a la password, i es guardarà juntament amb ella en la base de dades. Així s'evita que dos passwords iguals generin el mateix hash i els problemes que això comporta, com per exemple, atac per força bruta a totes les passwords del sistema alhora. Un altre atac relacionat és el de Rainbow table (taula arc de Sant Martí), que són taules d'associacions entre textos i el seu hash associat, per a evitar el seu càlcul i accelerar la cerca de la password, amb el salt, s'afegeix un grau de complexitat que evita que el hash associat a una password sigui únic.

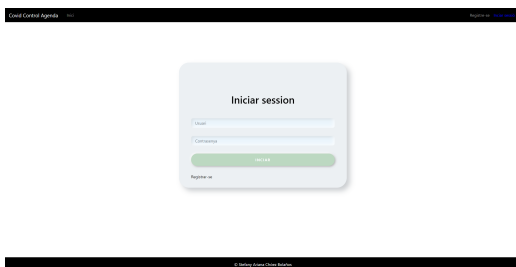


Fig. 4: Mòdul d'inici de sessió

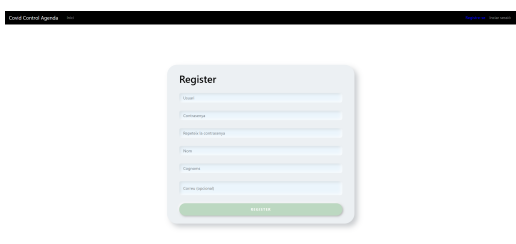


Fig. 5: Mòdul de registre

### 6.2.B Mòdul perfil usuari

En aquest mòdul l'usuari podrà visualitzar les seves dades i modificar-les si calgués, es fa una crida a l'API per obtenir les dades de l'usuari en concret i si l'usuari volgués modificar alguna de les dades, es realitzaria una crida per actualitzar les dades de l'usuari.

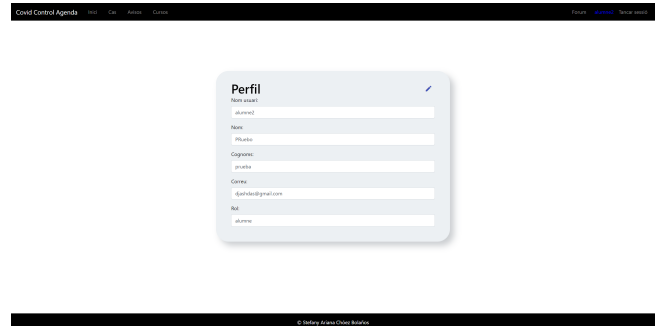


Fig. 6: Mòdul perfil

### 6.2.C Mòdul creació cas i visualització

Aquest és un dels mòduls claus de l'aplicació web, ja que aquesta aplicació vol ajudar a tenir un control sobre els casos de Covid-19 en els centres educatius, per tal de fer-ho s'haurà d'omplir un formulari amb tota la informació que altres usuaris podran visualitzar posteriorment. Un dels aspectes a destacar d'aquest mòdul és que segons el tipus d'usuari podrà realitzar determinades accions, per exemple, el professorat és l'únic que pot crear casos i, a més a més, el professorat en la seva pantalla de visualització podrà editar o eliminar el cas creat.

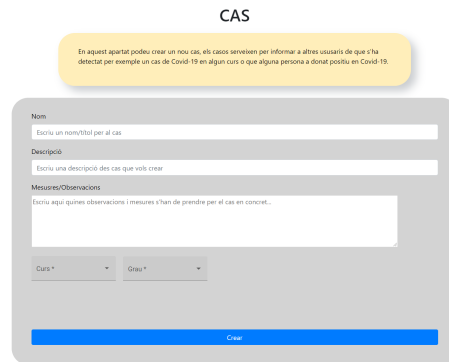


Fig. 7: Mòdul creació cas

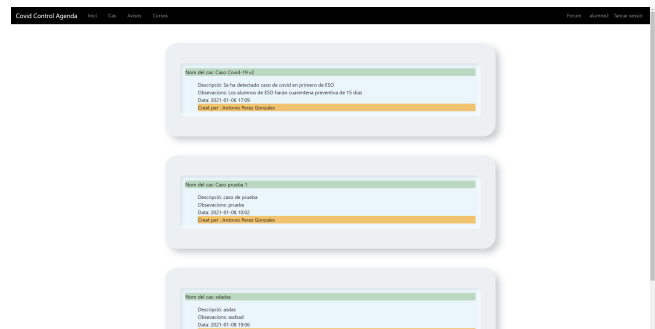


Fig. 8: Mòdul visualització cas

### 6.2.D Mòdul creació avisos i visualització

Aquest és l'altre mòdul clau de l'aplicació web, ja que aquesta aplicació a part de la creació de casos, el que es vol aconseguir amb aquesta aplicació és poder tenir un control pel que fa a l'organització per tal d'evitar aglomeracions als centres educatius, mitjançant aquest mòdul es podrà avisar tant als alumnes com als pares/tutors per on surten o han d'entrar els nens, avisar si hi ha qualsevol modificació o cancel·lació de classes, etc. Un dels aspectes a destacar d'aquest mòdul és que segons el tipus d'usuari podrà realitzar determinades accions, per exemple, el professorat és l'únic que pot crear casos i, a més a més, el professorat en la seva pantalla de visualització podrà editar o eliminar el cas creat.



Fig. 9: Mòdul creació avis

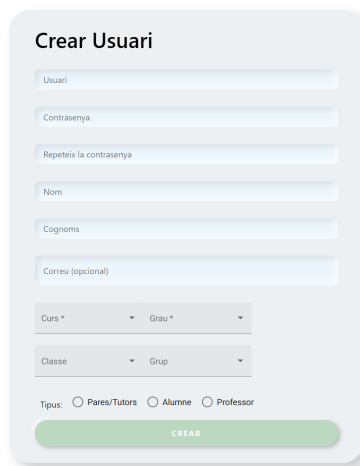


Fig. 11: Mòdul visualització avis

### 6.2.F Mòdul Forum (post i comentaris)

Aquest mòdul serveix perquè els usuaris puguin comunicar-se entre ells, per comentar dubtes, etc.

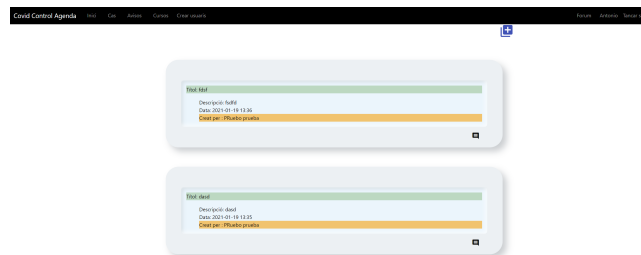


Fig. 12: Mòdul creació avis

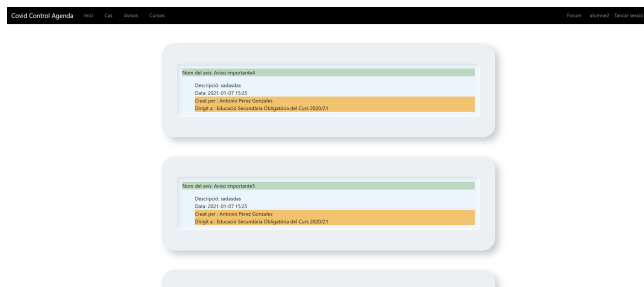


Fig. 10: Mòdul visualització avis

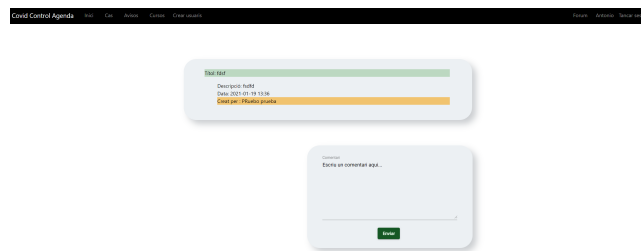


Fig. 13: Mòdul visualització avis

### 6.2.E Mòdul creació d'usuaris

Aquest mòdul serveix perquè el professorat pugui registrar als seus alumnes a l'aplicació, també es podrà registrar als pares/tutor, però l'aplicació ja dona l'opció que ells mateixos es puguin registrar a l'aplicació web mitjançant omplir un formulari.

## 7 TESTS

En aquest apartat s'explicaren tots les proves que s'han realitzat tant de l'API com de l'aplicació realitzada amb angular.

### 7.1 Test API

Com s'ha comentat en l'apartat de desenvolupament s'ha creat una API, d'aquesta API s'ha implementat les funcions CRUD per poder crear, llegir, actualitzar i eliminar les col·leccions de l'estructura de dades establerta. D'aquesta part de l'API s'ha realitzat un test per tal de garantir el correcte funcionament de l'API. Per poder explicar les proves realitzades i que no es faci tan extens l'explicació, s'explicarà una d'elles com a exemple, ja que les restes segueix la mateixa estructura de test i només canvia l'esquema que es vol testejar.

Prova de l'esquema USUARI:

- Test inserció d'usuari: es realitza una petició POST amb les dades de l'usuari que es vol introduir. Les dades per poder inserir un usuari són:
  - \* Nom
  - \* Cognoms
  - \* Usuari (nom d'usuari, exemple: StefanyChoez)
  - \* Contrasenya
  - \* Rol (si no s'introdueix, agafarà el valor per defecte que és el rol de user", aquest rol és el que s'assigna als pares/tutors)
  - \* Curs
  - \* Grau

```

Insercio usuari:
{
  ok: true,
  user: {
    rol: [ '5ff4386a026a6b77cc4d290' ],
    _id: '601d821707b0ff58dc03dfcd',
    usuari: 'test',
    correu: 'testmail@elmio.com',
    contrasenya: '$2a$08$x26gqhvjg1z9ttBH3wkjkoA9wLHRjikzbsf2e50vZKxqPiw2XkFwa',
    nom: 'test',
    cognoms: 'elmio',
    isLogged: false,
    _v: 1,
    curs: '5ff44b0319a09009d48f72d9',
    grau: '5ff44b9819a09009d48f72dc'
  }
}
√ Insertar un usuari correctament

```

Fig. 14: Test inserció usuari

- Test retornar tots els usuaris: es realitza una petició GET que ha de retornar tots els usuaris creats que hi ha en la base de dades.

```

get tots els usuaris:
{
  ok: true,
  user: [
    {
      rol: [],
      _id: '601d63cd73c0943908e187cc',
      usuari: 'sdfgfdsg',
      correu: 'prueba@gmail.com',
      contrasenya: '$2a$08$PwVcBLljqIscAPQXmXVBSuz6cCxnV83C6V7Q1Vyg.XeT1ERq123za',
      nom: 'user',
      cognoms: 'prueba',
      isLogged: false,
      _v: 0
    },
    {
      rol: [],
      _id: '601d63fd9b275846e0ad1f4f',
      usuari: 'sdfgfdsgsdfsddf',
      correu: 'prueba@gmail.com',
      contrasenya: '$2a$08$nz38MXrpf54puc9CZEDTE.Ov/mMnDRict0C0qQIO4CBZ.wbdXX1Pa',
      nom: 'user',
      cognoms: 'prueba',
      isLogged: false,
      _v: 0
    },
    {
      rol: [Array],
      _id: '601d646b871cf944b4a9f9a5',
      usuari: 'userprueba',
      correu: 'prueba@gmail.com',
      contrasenya: '$2a$08$kpA5w08BF52jd2HU/OPZQ.4yzfi20k42Vhqi01Cy8megSIVFUPYb2',
      nom: 'user',
      cognoms: 'prueba',
      isLogged: true,
      _v: 1,
      curs: '5ff44b0319a09009d48f72d9',
      grau: '5ff44b9819a09009d48f72dc'
    },
    {
      rol: [Array],
      _id: '601d821707b0ff58dc03dfcd',
      usuari: 'test',
      correu: 'testmail@elmio.com',
      contrasenya: '$2a$08$x26gqhvjg1z9ttBH3wkjkoA9wLHRjikzbsf2e50vZKxqPiw2XkFwa',
      nom: 'test',
      cognoms: 'elmio',
      isLogged: false,
      _v: 1,
      curs: '5ff44b0319a09009d48f72d9',
      grau: '5ff44b9819a09009d48f72dc'
    }
  ],
  total: 4
}
√ Retornar tots el usuaris creats

```

Fig. 15: Test retornar tots els usuaris

- Test retornar un usuari segons la seva ID: es realitza una petició GET que ha de retornar un usuari en concret mitjançant la seva ID.

```

get usuari amb el id 601d646b871cf944b4a9f9a5:
{
  rol: [ '5ff4386a026a6b77cc4d290' ],
  _id: '601d646b871cf944b4a9f9a5',
  usuari: 'userprueba',
  correu: 'prueba@gmail.com',
  contrasenya: '$2a$08$kpA5w08BF52jd2HU/OPZQ.4yzfi20k42Vhqi01Cy8megSIVFUPYb2',
  nom: 'user',
  cognoms: 'prueba',
  isLogged: true,
  _v: 1,
  curs: '5ff44b0319a09009d48f72d9',
  grau: '5ff44b9819a09009d48f72dc'
}
√ Retorna usuari amb id 601d646b871cf944b4a9f9a5

```

Fig. 16: Test retornar tots els usuaris

- Test modificar usuari: es realitza una petició PUT i es modifica el camp que es vulgui modificar en aquest exemple s'ha modificat el nom de l'usuari.

```

Modificació del nom usuari
{
  ok: true,
  message: 'Usuari modificat correctament',
  updateUser: { nom: 'test2' }
}
√ Modificar usuari

```

Fig. 17: Test retornar tots els usuaris

- Test eliminar usuari: es realitza una petició DELETE i s'elimina l'usuari segons la seva ID.

```

Eliminar usuari:
{
  rol: [],
  _id: '601d63cd73c0943908e187cc',
  usuari: 'sdfgfdsg',
  correu: 'prueba@gmail.com',
  contrasenya: '$2a$08$PwVcBLlJqISCAPQXMXVBSuz6CxnV83C6V7QlVYg.XeT1ERqj23za',
  nom: 'user',
  cognoms: 'prueba',
  isLogged: false,
  _v: 0
},
{
  rol: [],
  _id: '601d63fd9b275846e0ad1f4f',
  usuari: 'sdfgfdsgsdfsdff',
  correu: 'prueba@gmail.com',
  contrasenya: '$2a$08$nz38MXRpf54puc9CEDTE.Ov/mMnDRiCt0C0qQI04CBZ.wbdXlPa',
  nom: 'test2',
  cognoms: 'prueba',
  isLogged: false,
  _v: 0
},
{
  rol: [ '5ff4386a026a6b77cc4d290' ],
  _id: '601d646b871cf944b4a9f9a5',
  usuari: 'userprueba',
  correu: 'prueba@gmail.com',
  contrasenya: '$2a$08$kPA5w08BF52jd2HU/0PZQ.4yzfi2ok42VhQio1CyBmegSIVFUPYb2',
  nom: 'user',
  cognoms: 'prueba',
  isLogged: true,
  _v: 1,
  curs: '5ff44b0319a09009d48f72d9',
  grau: '5ff44b9819a09009d48f72dc'
},
{
  rol: [ '5ff4386a026a6b77cc4d290' ],
  _id: '601d82170b0ff58dc03dfcd',
  usuari: 'test',
  correu: 'testemail@elmio.com',
  contrasenya: '$2a$08$x26gqhvjjz9tt8H3wkj0A9wLHRjikzbsf2e50vXkqpiw2XkFwa',
  nom: 'test',
  cognoms: 'elmio',
  isLogged: false,
  _v: 1,
  curs: '5ff44b0319a09009d48f72d9',
  grau: '5ff44b9819a09009d48f72dc'
}
]
ok: true,
message: 'Usuari eliminat correctament',
_id: '601d82170b0ff58dc03dfcd'
}

{
  rol: [],
  _id: '601d63cd73c0943908e187cc',
  usuari: 'sdfgfdsg',
  correu: 'prueba@gmail.com',
  contrasenya: '$2a$08$PwVcBLlJqISCAPQXMXVBSuz6CxnV83C6V7QlVYg.XeT1ERqj23za',
  nom: 'user',
  cognoms: 'prueba',
  isLogged: false,
  _v: 0
},
{
  rol: [],
  _id: '601d63fd9b275846e0ad1f4f',
  usuari: 'sdfgfdsgsdfsdff',
  correu: 'prueba@gmail.com',
  contrasenya: '$2a$08$nz38MXRpf54puc9CEDTE.Ov/mMnDRiCt0C0qQI04CBZ.wbdXlPa',
  nom: 'test2',
  cognoms: 'prueba',
  isLogged: false,
  _v: 0
},
{
  rol: [ '5ff4386a026a6b77cc4d290' ],
  _id: '601d646b871cf944b4a9f9a5',
  usuari: 'userprueba',
  correu: 'prueba@gmail.com',
  contrasenya: '$2a$08$kPA5w08BF52jd2HU/0PZQ.4yzfi2ok42VhQio1CyBmegSIVFUPYb2',
  nom: 'user',
  cognoms: 'prueba',
  isLogged: true,
  _v: 1,
  curs: '5ff44b0319a09009d48f72d9',
  grau: '5ff44b9819a09009d48f72dc'
}
]
✓ Eliminar usuari primera posició

5 passing (50ms)

```

Fig. 18: Test eliminar usuari

## 7.2 Test aplicació

Per altra banda, per la part de l'aplicació que s'ha realitzat amb angular un cop s'ha realitzat el testatge del backend(l'API) es passa a la següent fase de testatge per la part de front-end. Igual que en l'apartat anterior poder explicar les proves realitzades i que no es faci tan extens l'explicació, s'explicarà una d'elles com a exemple, ja que les restes segueix la mateixa estructura de test i només canvia l'esquema que es vol testejar.

Prova del mòdul de registre:

- Registre Component test aïllat: es comprova la lògica del component.

- \* Component s'hagi creat correctament
- \* Estat inicial del component correcte
- \* La variable submitted ha de ser true quan registre
- \* El valor del formulari s'ha d'actualitzar quan canvia l'input
- \* El formulari no vàlid deure ser verdader quan el formulari no és vàlid

- Registre Component test superficial: es comprova que es mostrin els missatges d'error i que en fer clic s'activi la crida a la funció.

- \* Crear un formulari amb l'entrada usuari, contrasenya, repeteix contrasenya, nom, cognoms, correu i el botó de registre.
- \* Mostrar missatge error de nom d'usuari quan el nom d'usuari és buit
- \* Mostrar missatge error de contrasenya quan el camp de contrasenya és buit
- \* Mostrar missatge error de correu quan el camp de correu és buit
- \* Mostrar missatge error de confirmació de la contrasenya quan el camp de repeteix contrasenya és buit
- \* Crear un formulari amb la entrada usuari, contrasenya, repeteix contrasenya, nom, cognoms, correu i el botó de registre
- \* Mostrar missatge error de nom quan el camp de nom es buit
- \* Mostrar missatge error de nom d'usuari quan el nom de usuari es buit
- \* Mostrar missatge error de nom usuari i contrasenya quan tots dos camps estan en blanc
- \* Mostrar el missatge error de nom i cognom quan tots dos camps estan en blanc
- \* Mostrar el missatge error de nom i correu quan tots dos camps estan en blanc

```

Registre Component Test superficial
• Mostrar el missatge error de nom i correu quan tots dos camps estan en blanc
• Mostrar missatge error de cognoms quan el camp de cognoms es buit
• Mostrar missatge error de contrasenya quan el camp de contrasenya es buit
• Mostrar el missatge error de nom i cognom quan tots dos camps estan en blanc
• Mostrar missatge error de correu quan el camp de correu es buit
• Mostrar missatge error de confirmació de la contrasenya quan el camp de repeteix contrasenya es buit
• Crear un formulari amb la entrada usuari, contrasenya, repeteix contrasenya, nom, cognoms, correu i el botó de registre
• Mostrar missatge error de nom quan el camp de nom es buit
• Mostrar missatge error de nom d'usuari quan el nom de usuari es buit
• Mostrar el missatge error de nom usuari i contrasenya quan tots dos camps estan en blanc

Registre Component Test aïllat
• Component creat correctament
• El valor del formulari ha de actualitzar-se quan canvia el input
• El formulari no vàlid deure ser verdader quan el formulari no és vàlid
• Estat inicial del component
• La variable submitted ha de ser true quan registre

```

Fig. 19: Test front-end mòdul registre



## 8 RESULTATS

Respecte als resultats obtinguts, segons els objectius que es van plantejar per al projecte s'han completat tots els que tenien prioritats alta, però per falta de temps no s'ha pogut realitzar la funcionalitat extra que s'havia plantejat a l'inici del projecte, però que en un futur si es vol continuar desenvolupant aquesta aplicació, es podria acabar d'implementar aquesta funcionalitat extra. Per altra banda, s'ha pogut posar a familiars davant de l'aplicació i deixar-los que naveguin per tal de tenir un feedback, i com a resultat a estat bastant bo, ja que la majoria ha dit que és bastant senzill d'utilitzar i que al ser simple pot afavorir al fet que la gent gran s'orienti millor. Per altra banda, també m'han comentat que la idea és bona per als temps que estem vivint. No s'ha volgut fer un test a les persones, ja que a vegades la gent contesta el test molt ràpid per acabar abans i s'ha preferit que ells mateixos diguin el que li ha semblat l'aplicació.

## 9 CONCLUSIONS

Segons els que es va plantejar a l'inici del projecte, s'ha pogut fer totes les funcionalitats que l'aplicació hauria de tenir menys la part extra que es va afegir a l'inici per donar més valor a l'aplicació però que si no es feia es deixaria per una futura ampliació. Com a futures ampliacions també es podria fer:

- Millorar el sistema de creació casos y avisos per tal de poder crear grups de persones a les que es vulgui enviar el cas o els avisos.
- Implementar sistema de notificacions personalitzat que només el rebien les persones a les que vagi dirigit el cas o l'avís.

## 10 AGRAÏMENTS

En primer lloc, agraeixo al meu tutor de projecte Marc Talló la disposició i llibertat donada a l'hora de planificar i desenvolupar el treball de forma autònoma. Per altra banda també agraeixo a la meua família per invertir part del seu temps en provar l'aplicació i donar-me feedback.

## REFERÈNCIES

- [1] Getting started with a basic Angular app, 2020, Equip d'enginyer d'angular, <https://angular.io/start>
- [2] Cómo crear un API REST usando Node.js, Express y MongoDB, 21 de setembre 2017, Carlos Azaustre, <https://carlosazaustre.es/como-crear-una-api-rest-usando-node-js>
- [3] Estructura de una API Rest con NodeJS, Express y MongoDB, 19 de febrer 2018, William Bastidas, <https://medium.com/williambastidasblog/estructura-de-una-api-rest-con-nodejs-express-y-mongodb-cdd97637b18b>
- [4] Ejemplo API REST en express.js y mongoddb, 21 de maig 2019, David Poza Suárez, <https://davidinformatico.com/ejemplo-api-express-js-mongodb/>
- [5] PWA, Entre la web y las apps con Angular, 3 d'octubre 2019, Alberto Basalo, <https://academia-binaria.com/pwa-entre-la-web-y-las-apps-con-angular/>
- [6] PWA y Services Worker en Angular, 17 d'octubre 2017, Enrique Rubio, <https://www.adictosaltrabajo.com/2017/10/17/pwa-y-services-worker-en-angular/>
- [7] Getting Started, 2020, Col·laboradors mongoose, <https://mongoosejs.com/docs/index.html>
- [8] Map Set in Typescript (Angular), 18 de gener 2019, Sathish kumar Ramalingam, <https://medium.com/ramsatt/map-set-in-typescript-angular-643e506e6c3c>
- [9] Angular 9—8 Owl Datepicker Timepicker using Ng Pick Datetime Tutorial by Example, 8 de maig 2020, Jolly.Exe, <https://www.freakyjolly.com/angular-datepicker-timepicker-using-ng-pick-datetime-tutorial-by-example/#.X94eV9hKibg>
- [10] Angular 10—9—8 NgSelect with Single/Multiple Selection and Search Filter, 22 d'agost 2020, Jolly.Exe, <https://www.freakyjolly.com/angular-ngselect-with-single-multiple-and-search-filter-tutorial/#.X94edthKibh>
- [11] Angular Drop Down Example, 2020, Infragistics, <https://www.infragistics.com/products/ignite-ui-angular/angular/components/drop-down>
- [12] Data Binding in Angular DropDownList component, 2020, Syncfusion, <https://ej2.syncfusion.com/angular/documentation/drop-down-list/data-binding/#data-binding>

- [13] Angular expansion list, 2020, stackblitz, <https://stackblitz.com/edit/angular-expansion-list?file=app%2Fexpansion-list%2Fexpansion-list.component.ts>
- [14] HttpClient VS Http – El nuevo servicio http de Angular, 14 de juliol 2020, Techiediaries Team, <https://www.techiediaries.com/angular-http-client/>
- [15] Angular 10 HttpClient , 15 de novembre 2017, Enrique Oriol, <http://blog.enriqueoriol.com/2017/11/httpclient-vs-http-angular.html>

## APÈNDIX

### A.1 Diagrama de cas d'ús

En la figura que es mostra a continuació, es pot observar el diagrama de cas d'ús amb totes les funcionalitats que ha de tenir l'aplicació.

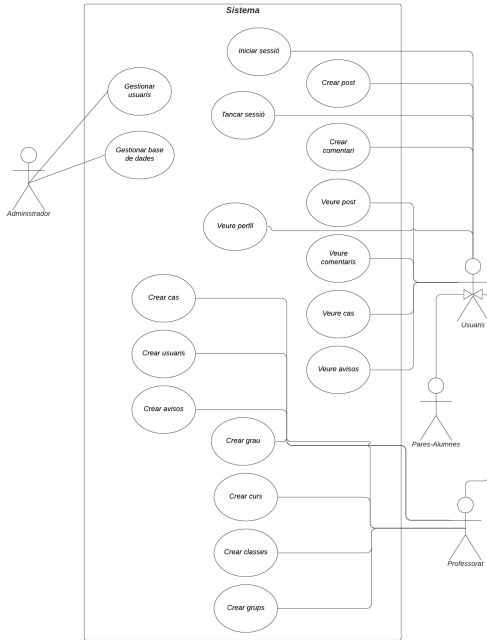


Fig. 20: Diagrama de cas d'ús

### A.2 Diagrama de seqüència registre

En la figura següent es pot observar el diagrama de seqüència que simula tot el procés que es realitza quan es vulgui registre un usuari.

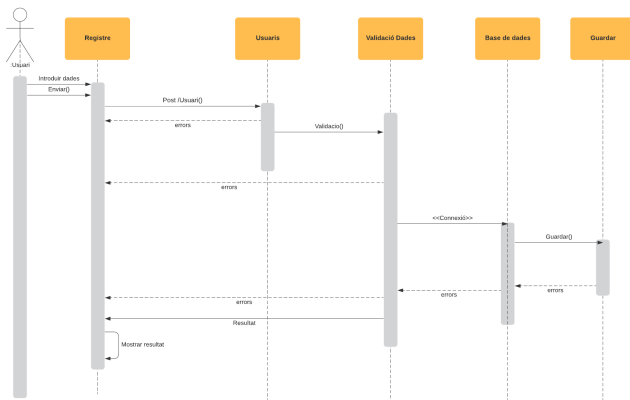


Fig. 21: Diagrama de seqüència registre

### A.3 Diagrama d'activitat get cas

En la figura que es mostra a continuació, es pot observar la representació del procés que es realitza quan un usuari vol visualitzar la llista dels casos que s'han creat i el seu estat és actiu.

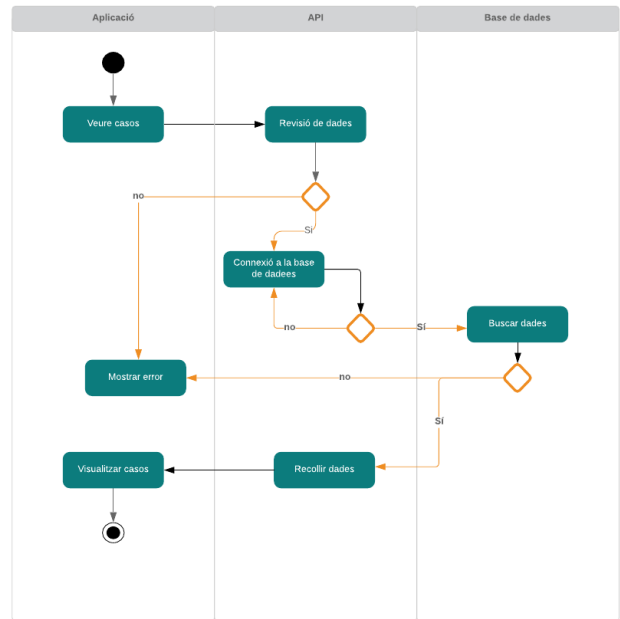


Fig. 22: Diagrama d'activitat get cas

### A.4 Interfície Cas

En la figura següent es pot observar el formulari que es mostra quan l'usuari del professorat vol crear un cas nou.

**CAS**

En aquest apartat podeu crear un nou cas, els casos serveixen per informar a altres usuaris de que s'ha detectat per exemple un cas de Covid-19 en algun curs o que alguna persona a donat positiu en Covid-19.

Nom

Descripció

Mesures/Observacions

Curs \*  Grau \*

Crear

Fig. 23: Interfície cas

### A.5 Interfície Avis

En la figura següent es pot observar el formulari que es mostra quan l'usuari del professorat vol crear un avís nou per informar els alumnes i pares d'algun canvi imprevist o planificat.



Fig. 24: Interfície avís

## A.8 Interfície Registre

En la figura següent es pot observar el formulari que es mostrarà quan els pares o familiars vulguin registre-se

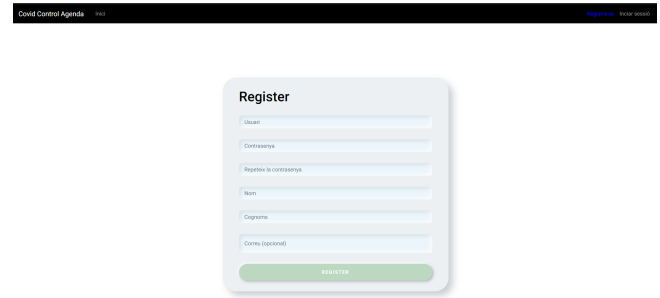


Fig. 27: Interfície registre

## A.6 Interfície Crear Usuari

En la figura següent es pot observar el formulari que s'ha d'omplir quan el professorat vulgui crear un nou usuari.

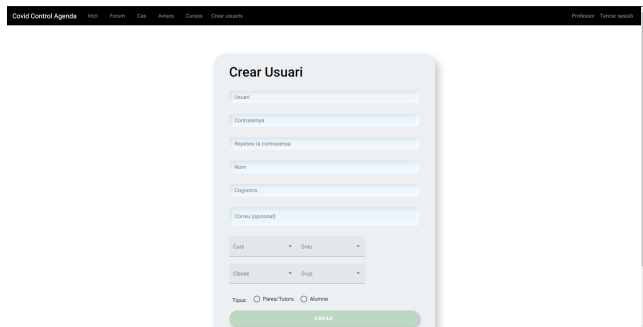


Fig. 25: Interfície creació usuari

## A.9 Interfície Curs/Grau/Classe/Grup

En les figures que es mostren a continuació, es pot observar els formularis que es mostren quan l'usuari del professorat vol crear un curs nou amb els seus graus, classes i grups.

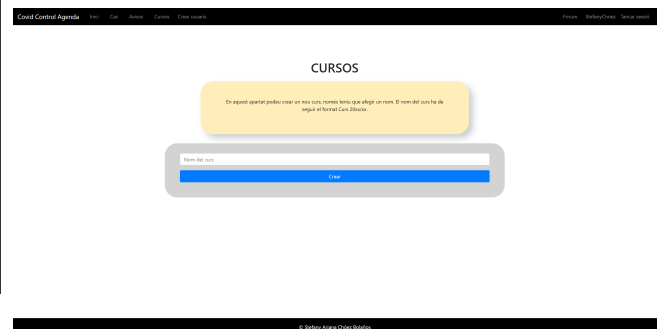


Fig. 28: Interfície curs

## A.7 Interfície Iniciar sessió

En la figura següent es pot observar el formulari que es mostrarà quan els usuaris vulguin iniciar sessió a l'aplicació.

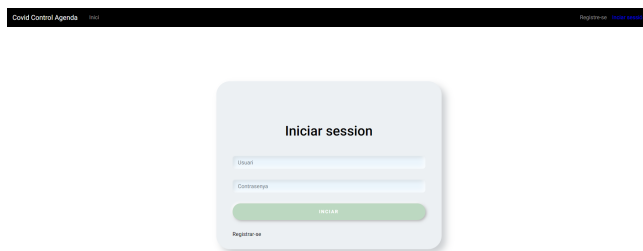


Fig. 26: Interfície iniciar sessió

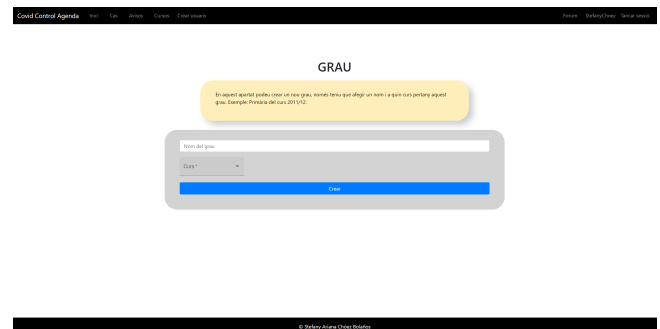


Fig. 29: Interfície grau



