PROJECTE CONNECTA COVID-19



Pla de desenvolupament software y documentació tècnica.

PROJECTE AVANÇAT D'ENGINYERIA

DESENVOLUPAT PER

Ignasi Agustí Bosch Auri Botines Puertas Enric Carrera Aguiar Sara Hortelano Lozano Jesús López Pocino Àngela Niubò Badia Alex Ruiz Gallart Marc Sarri Mcbride

DESCRIPCIÓ DEL DOCUMENT

Títol	Documentació tècnica del projecte "CONNECTA COVID-19".
Descripció	És un entorn software desarrollat per tal de facilitar la comunicació pacient-familiar en territori català durant el període de crisis sanitària provocada pel virus <i>Covid-19</i> , de manera que el familiar tingui una informació actualitzada de l'estat del pacient si aquest es troba dins de la xarxa d'hospitals catalans.
Autor	Membres de l'equip desenvolupador
Versió Android	0.1.6a (pass: zu6Ard)
Versió IOS	
Versió Web	1.0

ÍNDEX

DESCRIPCIÓ DEL DOCUMENT	2
ÍNDEX	2
INTRODUCCIÓ	4
PROPÒSITS I OBJECTIUS	4
REQUERIMENTS DEL PROJECTE	5
ARQUITECTURA DEL SISTEMA	6
DISSENY DE L'ARQUITECTURA CLIENT-SERVIDOR	6
Descripció del sistema	6
Elements del sistema	6
Infraestructura	6
Tipus de clients	6
ESTRUCTURA I DISSENY DE LA BBD	7
ESTRUCTURA DE LA BBDD	7
API D'ACCÉS REMOT	10
DISSENY I IMPLEMENTACIÓ DE CLIENT FAMILIAR	21
ANDROID	21
Screenflow	21
Detalls de les pantalles	22
Estructura i codi de l'aplicació	26
Crides remotes	27
Tractament de dades rebudes	28
Recursos de l'App	28
Aspectes de seguretat	29
IOS	30
Screenflow	30
Detalls de les pantalles	31
Estructura i codi de l'aplicació	38
Elements de seguretat	39
Dades persistents de l'aplicació	39
Crides remotes	39

Tractament de dades rebudes	39
Recursos de l'App	40
WEB	42
Detalls de Pantalles	42
Estructura i codi de l'aplicació	46
Recursos	48
Seguretat	48
DISSENY I IMPLEMENTACIÓ DE LA PART PEL PERSONAL MÈDIC	49
Detalls de Pantalles	49
Estructura i codi de l'aplicació	55
Recursos	61
Seguretat	62

1. INTRODUCCIÓ

Aquest document té com a objectiu explicar tot el procés dut a terme durant el desenvolupament de l'aplicació Connecta Covid 19(CC19) sol·licitada per el Institut Català de la Salut(ICS) a la Universitat Politècnica de Catalunya.

A través dels diferents apartats desglossarem el projecte per tenir una visió mes exacte i detallada de com s'ha aconseguit assolir una primera versió estable i preparada per sotmetre's a una fase de test en un entorn real.

Pel que fa a l'equip desenvolupador, a l'inici de l'assignatura de PAE(Projecte Avançat d'Enginyeria) estàvem involucrats en el desenvolupament d'una APP per Android i iOS per la corporació TRESC: Comunitat de Cultura. I no va ser fins a l'inici de l'estat d'alarma i amb la sol·licitud del ICS, que donada la naturalesa del projecte que realitzàvem, se'ns va oferir la possibilitat de reorganitzar el nostre projecte en la APP CC19.

Donada la situació excepcional, vam veure la realització de CC19 com la manera d'afegir el nostre granet d'arròs en la lluita contra la Covid i els seus efectes negatius en la societat. Per tant no vam dubtar a l'hora d'acceptar el repte i entregar-nos 100% per assolir un desenllaç satisfactori.

També és obligatori remarcar l'ajuda rebuda des del conjunt UPC, ja sigui des del nostre tutor de PAE, JUAN LUIS GORRICHO, el director de l'escola ETSETB, JOSEP PEGUEROLES i al Personal d'Administració i Serveis, GABRIEL VERDEJO i FERNANDO GALINDO, sense els quals no ho haguéssim aconseguit.

1.1. PROPÒSITS I OBJECTIUS

A causa de les estrictes mesures de prevenció imposades als hospitals per combatre la Covid qualsevol mena de visita als centres mèdics quedava prohibida, generant que pacients ingressats i els seus familiars i/o persones del seu entorn es trobessin totalment incomunicats. A part, en els casos més greus, a causa del seu estat de salut, aquests pacients no poden utilitzar ni el seu telèfon mòbil ni cap eina de comunicació.

Tota aquesta problemàtica feia sorgiren el pacient un sentiment de solitud i per les persones del seu entorn angoixa e incertesa de no saber amb exactitud el seu estat.

I es aquí on neix la necessitat de la creació de Connecta Covid 19(CC19).Un sistema de comunicació digital (web, app) fàcilment actualitzable per part dels professionals sanitaris per informar de forma permanent a les famílies que no poden acompanyar als pacients ingressats.

L'objectiu principal és de facilitar als interessats informació del pacient en conceptes simples, com per exemple el seu estat, la localització i les dates dels esdeveniments. I així alleugerir la preocupació.

L'altre beneficiari de CC19 és el personal mèdic, altament col·lapsat e incapaç de donar l'abast a l'hora de comunicar de forma personalitzada pacients i famílies. Una comunicació que respon a una casuística molt gran depenent dels centres, però en molts casos es centra en una trucada cada 24 hores.

1.2. REQUERIMENTS DEL PROJECTE

Assolir els objectius és vital, i per això la creació d'un canal de comunicació Pacient<->Família al qual el personal mèdic pugui utilitzar de forma rápida e intuitiva.

Pel que fa la versió familiar, la APP creada ha de facilitar la informació necesaria del pacient una vegada s'hagin introduït les seves dades de seguiment de manera correcta. Aquestes dades són el CIP(número de la targeta CATSALUT), el primer cognom del pacient i un número Identificador que serà facilitat en el moment d'ingrés del pacient. Aquestes dades permeten asegurar la privacitat del pacient, ja que solament persones pròximes tenen la capacitat d'obtenir aquestes dades, i també el funcionament independent del sistema.

I per la part orientada al personal mèdic (professional), l'eina de comunicació ha de permetre descongestionar i agilitzar la tasca d'actualització dels estats dels seus pacients. Amb un disseny amigable e intuïtiu, reduint el màxim possible la redundancia, ja que hem de tenir en compte que un treballador repeteix els passos vàries vegades en un torn.

2. ARQUITECTURA DEL SISTEMA

2.1. DISSENY DE L'ARQUITECTURA CLIENT-SERVIDOR

- 2.1.1. Descripció del sistema
- 2.1.2. Elements del sistema
- 2.1.3. Infraestructura
- 2.1.4. Tipus de clients

3. ESTRUCTURA I DISSENY DE LA BBD

En aquest apartat definim l'estructura de la base de dades (Data Base, DB), la qual conté tota la informació gestionada per l'aplicació Connecta Covid 19 (CC19). A part també definirem tot el que és necessari per realitzar consultes i tractar les respostes de les API.

3.1. ESTRUCTURA DE LA BBDD

La DB emprada ubicada en el servidor del ICS està formada per un total de quatre taules, Pacients, Estats, Fitxes i Prof (per al personal mèdic). La definició d'aquestes taules és la següent:



PACIENTS: La taula pacients conte les dades principals de cada pacient. Aquestes dades són utilitzades per validar l'existència d'un pacient en concret en la DB i a posteriori permet accedir als Estats i/o la Fitxa d'aquest pacient. Les dades principalment usades són:

ID	CODI_CENTRE	CIP	NHC_EPISODI	NOM	COGNOM1	COGNOM2	DATA
1	C_01	CIP_01	000000001000000001	NOM_01	COGNOM_1_01	COGNOM_2_01	2020-05-15 08:32:13

 ID: Variable primària propia del servidor per comptabilitzar el nombre total de files de la taula.
 *D'ara en endavant s'omitira aquesta variable ja que està present en

totes les taules de la DB i la seva funció no varia.*

- Codi_Centre: Codi del centre en el qual el pacient està ingressat.
 Essencial per poder confeccionar la llista total de pacients en l'aplicatiu per personal mèdic.(p.e. Vall d'Hebron -> VH)
- CIP: Codi d'identificació personal de la Targeta sanitària individual del pacient. Essencial per poder validar l'existència del pacient en la DB.
- NHC_Episodi: Conjunt de 20 xifres format a partir del NHC (Número d'Historial Clínic) i el número d'episodi actual. A aquest conjunt també és anomenat Identificador de seguiment. Essencial per poder validar l'existència del pacient en la DB.

- Nom, Congnom1/2: Credencials del pacient. Essencials per poder validar l'existència del pacient en la DB.
- DATA: Data en la qual el pacient ha estat inserit en la taula Pacients.

ESTATS: Com bé indica el nom, la taula conte TOTS els estats associats als Identificadors (NHC_Episodi) continguts en Pacients. Donat un Identificador, permet seleccionar els estats solament d'aquest Identificador. Els estats d'un Identificador són additius, cada estat nou representa una nova entrada a la taula.Les dades principalment usades són:

NHC_EPISODI	ESTAT	COMENTARI	DATA
000000010000000010	0	NULL	2020-05-15 08:32:13
000000010000000000	90	NULL	2020-05-10 06:00:00
000000010000000010	91	NULL	2020-05-15 16:00:00

- NHC_Episodi: Conjunt de 20 xifres format a partir del NHC (Número d'Historial Clínic) i el número d'episodi actual. A aquest conjunt també és anomenat Identificador de seguiment. Essencial per poder validar l'existència del pacient en la DB.
- ESTAT: Variable usada per definir un estat propi prèviament definit en l'entorn de l'aplicatiu. Aquest estat té un valor numèric que correspon a una frase que defineix l'estat. (p.e. 2 -> Ingressat/da a planta).
 Els estats poden anar del 0 al 99, actualment n'hi han definits 13. Cal remarcar que els estats del 90 a 99 estan marcats com a Estats Finals i

remarcar que els estats del 90 a 99 estan marcats com a Estats Finals i per tant quan un pacient rebi un Estat Final, serà eliminat de la DB en un període de cinc dies.

- 0 Pendent de triatge
- 1 Atès/a a urgències
- 2 Ingressat/da a planta
- 3 Pendent de proves
- 4 Ingressat/da a UCI
- 5 Trasllat de la UCI per anar a planta
- 6 Us trucarem en els propers 120'
- 7 El metge/ssa ha fet valoració, situació estable
- 91 Previsió d'alta en les properes hores
- 9 Pendent de trasllat a un altre centre
- 92 Efectes personals entregats
- 11 Possibilitat d'enviar missatge personalitzat
- 12 Possibilitat de videoconferència avui

- COMENTARI: Camp buit que permet incloure observacions a l'hora d'incloure un estat.
- DATA: Data en la qual l'estat ha estat inserit en la taula Estats. Permet ordenar els estats cronològicament i tenir control sobre quant de temps ha passat des de l'última actualització d'estat d'un pacient.

FITXES: En aquesta taula s'hi troba, la informació no medica relacionada amb l'entorn social de cada pacient contingut en la DB. Com en el cas d'Estats, donat un Identificador es pot accedir a la seva fitxa. Cada Identificador té una única fitxa a la taula.Les dades principalment usades són:

NHC_EPISODI	APELATIU	CONTACTES	PARENTIU	NOTES	LLIURE
000000001000000001	Conchi	Pep tlf: 6000067	Fill	Notes del dia 10 de maig	Bona Tarda

- NHC_Episodi: Conjunt de 20 xifres format a partir del NHC (Número d'Historial Clínic) i el número d'episodi actual. A aquest conjunt també és anomenat Identificador de seguiment. Essencial per poder validar l'existència del pacient en la DB.
- APELATIU: Apel·latiu informal del nom del pacient amb el qual s'ha sent còmode.
- CONTACTES: Persona/es amb les quals prioritzar el contacte en cas de necessitat.
- PARENTIU: Parentiu amb el pacient de la persona de contacte.
- NOTES: Observacions respecte al comportament/situació del pacient. (p.e. Dificultats a l'hora de menjar)
- LLIURE: Observacions respecte a l'entorn del pacient. (p.e. Conflicte amb el seus fills)

PROF: Taula troncal pel funcionament de la versió Professional enfocada al personal mèdic. Aquesta taula conte les credencials per validar els logins del personal i generar la llista completa dels pacients ingressats en l'hospital associat al personal que ha accedit al servei. Les dades principalment usades són:

USER	PASSWORD	CODI_CENTRE	DESC_CENTRE
C_01_user	C_01_pwd	C_01	CENTRE_01

- USER: Usuari registrat a la taula.
- PASSWORD: Contrasenya associada al usuari.
- Codi_Centre: Codi del centre en el qual el usuari(Personal Mèdic) hi treballa.(p.e. Vall d'Hebron -> VH). Permet obtenir tots els pacients de la

taula Pacients on el seu Codi_Centre sigui el mateix que el Codi_Centre associat al professional.

• Desc_Centre: Descripció més extensa sobre el centre sanitari.

3.2. API D'ACCÉS REMOT

En aquest apartat veurem com les API gestionen la informació introduïda en les APP's i la informació que s'ha de mostrar en les APP's. Les API utilitzen les variables esmentades en les taules de l'apartat anterior. Totes les API's parteixen de la mateixa URL: <u>https://wsm.ics.gencat.cat/connectaCovid/XXXXXX</u>. Totes les crides contenen els paràmetres 'usuari i password' establerts exclusivament per validar que les crides a les API les realitzem nosaltres i evitar ingerències exteriors.

En total tenim cinc API's: apiFa, apiPacientsFitxa, apiPacientsLLoc, apiAddEstat, apiAddFitxa.

API ENDPOINT	apiFa.php				
DESCRIPCIÓ	Mostra l'histor credencials en	Mostra l'historial d'estats del pacient el qual s'han enviat les credencials en la consulta.			
VERSIÓ	Usada per la v	ersió Familia	ar		
MÈTODE	POST				
PARÀMETRES REQUEST	Els dispositius Paràmetre Identificado r CIP	mòbils fan s TIPUS {longitud } String {20} String {10}	ervir JSON i la WEB form-data. Descripció NHC_Episodi únic per pacient. Format per 20 números. CIP personal del pacient. Format per 4 majúscules al inici i 6 números al final.		
	Cognom1	String	Primer Cognom del Pacient		
	usuari	String	Paràmetre definit per validar la connexió a la api		

	password	String	Paràmetre definit per validar la connexió a la api			
RESPONSE	Paràmetre Descripció					
			· · · · · · · · · · · · · · · · · · ·			
	status	'Error'=Pacie	cient No trobat a la DB			
	dades{}	JSON Arra informaci	ay que conté la ó dels estats del pacient.			
	dades.ID	Contingut pacient o fa servir e NHC_Epis	Contingut en dades{}. És el CIP del pacient o si el pacient no te CIP, es fa servir el NHC Episodi(Identificador)			
	dades.Estats []	Contingut que conté estat del F	Contingut en dades{}. JSON Array que conté Estat,Lloc,Data per cada estat del Pacient.			
	dades.Estats [].Estat	Valor num	èric del estat.			
	dades.Estats [].Lloc	Centre des de el qual s'ha actualitzat aquest estat.				
	dades.Estats [].Data	Data del n nou estat.	noment de la inserció del			
EXEMPLE	REQUEST: { "Identificador":"0000000000000000116", "CIP":"POCA000000004", "Cognom1":"Politècnica", "usuari":"abc1234", "password":"pwd" } RESPONSE (Pacient Trobat) { status": "OK", "dades": { "ID": "00000000000000116",					

"Estats": [
{
"Estat": "0",
"Lloc": "Hospital Santa Creu de Jesús",
"Data": "2020-05-08 13:04:07"
}
]
}
}
RESPONSE (Pacient no trobat)
{
"status": "ERROR",
"dades": "Not found."
}

API ENDPOINT	apiPacientFitxa.php			
DESCRIPCIÓ	Entrega les dades necessàries per completar l'Historial d'un pacient accessible des de la TaulaProfessional d'un hospital. Retorna l'historial d'estats i la fitxa.			
VERSIÓ	Usada per la v	versió Profes	sional.	
MÈTODE	POST			
PARÀMETRES REQUEST	Format formulari HTML(form-data). Mateixos paràmetres que per la apiFa. També accepta format JSON.			
	Paràmetre	TIPUS {longitud }	Descripció	
	Identificad or	String {20}	NHC_Episodi únic per pacient. Format per 20 números.	
	CIP	String {10}	CIP personal del pacient. Format per 4 majúscules al inici i 6 números al final.	
	Cognom1	String	Primer Cognom del Pacient	

	usuari	String	Paràmetre definit per validar la connexió a la api		
	password	String	Paràmetre definit per validar la connexió a la api		
PARÀMETRES RESPONSE	Retorna els mateixos paràmetres que la apiFa afegint la fitxa del pacient. Objecte JSON amb la següent estructura:				
	Paràmetre	Descripci	ó		
	status	'OK'=Pac 'Error'=Pa	ient Trobat acient No trobat a la DB		
	dades{}	JSON Arr estats, l'h de la seva	ray que conté la informació dels nistorial del pacient i les variables a fitxa.		
	dades.ID	Contingu si el pacie NHC_Epis	Contingut en dades{}. És el CIP del pacient o si el pacient no te CIP, es fa servir el NHC_Episodi(Identificador)		
	dades.Estats []	Contingut en dades{}. JSON Array que conté Estat,Lloc,Data per cada estat del Pacient.			
	dades.Estats [].Estat	Valor numèric del estat.			
	dades.Estats [].Lloc	Centre des de el qual s'ha actualitzat aquest estat.			
	dades.Estats [].Data	Data del moment de la inserció del nou estat.			
	dades.Fitxa[]	Contingu les variat	t en dades{}. JSON Array que conté les de la fitxa del pacient consultat.		
	dades.Fitxa[] .Parentiu	Parentiu	de la persona/es de contacte.		
	dades.Fitxa[] .Apelatiu	Apel·latiu	ı del pacient		
	dades.Fitxa[] .Contactes	Prioritzac	ió de contacte		
	dades.Fitxa[] .Notes	Camp d'c	bservacions		

r.

	dades.Fitxa[] Camp d'observacions extra . Lliure		
EXEMPLE	REQUEST: { "Identificador":"0000000000000000116", "CIP":"POCA000000004", "Cognom1":"Politècnica", "usuari":"abc1234", "password":"pwd"		
	<pre>} RESPONSE (Pacient Trobat) { "status": "OK", "dades": { "ID": "POCA000000004", "Estats": [{ "Estats": [{ "Estat": "0", "Lloc": "Hospital Santa Creu de Jesús", "Data": "2020-05-08 13:04:07" }, {</pre>		
	"Lloc": "Hospital Santa Creu de Jesús", "Data": "2020-05-11 16:24:34" }], "Fitxa": { "Parentiu": "Filla", "Apelatiu": "Carmeta", "Contactes": "Maria - tlf: 898989892", "Notes": "Li agraden les flors", "Lliure": ""		

}
}
}
RESPONSE (Pacient no trobat)
"status": "ERROR",
"dades": "Not found."
}

API ENDPOINT	apiPacientsLloc.php		
DESCRIPCIÓ	Permet obtenir el llistat e informació dels pacients ingressats a l'hospital associat a les credencials del professional. Per cada pacient de la llista només retorna el últim estat.		
VERSIÓ	Usada per la v	versió Profes	sional.
MÈTODE	POST		
PARÀMETRES REQUEST	Format formulari HTML(form-data). També accepta format JSON.		
	Paràmetre	TIPUS {longitud}	Descripció
	user	String	Nom d'usuari del professional
	pwd	String	Contrasenya associada al usuari
	usuari	String	Paràmetre definit per validar la connexió a la api
	password	String	Paràmetre definit per validar la connexió a la api
PARÀMETRES RESPONSE	Objecte JSON amb la següent estructura:		
	Paràmetre	iràmetre Descripció	
	status	'OK'=Usua	ri Trobat

		'Error'=Usuari No trobat a la DB
	dades[]	JSON Array que conté la informació dels pacients ingressats al hospital associat al professional
	dades[]. Identificado r	Contingut en dades[]. És el Identificador(NHC_Epi) del pacient
	dades[].CIP	CIP personal del pacient.
	dades[].No m	Nom del pacient
	dades[]. Cognom1	Primer Cognom del pacient
	dades[]. Cognom2	Segon Cognom del pacient
	dades[].Esta t	Estat actual del pacient. Aquesta API només retorna el últim estat del pacient
	dades[].Dat a	Data del estat actual
	dades[].Lloc	LLoc de l'estat actual
	dades[].Ulti m	Camp numèric que retorna en minuts la diferencia de la data entre la data del estat i el moment de fer la crida. Si aquest temps supera 24hores es retorna '+1 dia'.
EXEMPLE	REQUEST: { "user":"usua "pwd":"cont "usuari":"ab "password": }	ari", trasenya", c1234", "pwd"
	RESPONSE (Us {	uari Trobat)
	"status": "O	К",
	"dades": [{	

"Identificador": "00000000000000000116",
"CIP": "POCA000000004",
"Nom": "Universitat",
"cognom1": "Politècnica",
"cognom2": "de Catalunya",
"Estat": "0",
"Data": "2020-05-08 13:04:07",
"Lloc": "Hospital Comarcal d'Amposta",
"Ultim": "135"
},
{
"Identificador": "00000111110000000016",
"CIP": "GACA3333003333",
"Nom": "Nom" <i>,</i>
"cognom1": "Ga",
"cognom2": "Ca",
"Estat": "6" <i>,</i>
"Data": "2020-05-10 18:15:37",
"Lloc": "Hospital Comarcal d'Amposta",
"Ultim": "+1 dia"
}
],
}
"status": "ERROR",
"dades": "Not found."
1

API ENDPOINT	apiAddEstat.php
DESCRIPCIÓ	Permet afegir un nou estat a un pacient.
VERSIÓ	Usada per la versió Professional.

MÈTODE	POST		
PARÀMETRES REQUEST	Format formulari HTML(form-data). També accepta format JSON.		
	Paràmetre	TIPUS {longitud }	Descripció
	ldentificado r	String {20}	NHC_Episodi únic per pacient. Format per 20 números.
	Estat	0-99	Número associat al estat a afegir.
	Comentari	String	Comentari al estat.
	usuari	String	Paràmetre definit per validar la connexió a la api
	password	String	Paràmetre definit per validar la connexió a la api
PARÀMETRES RESPONSE	Objecte JSON Paràmetre	ecte JSON amb la següent estructura:	
	status	"OK" = S'ha "Error" = N	a afegit correctament l'estat o afegit correctament l'estat
	dades[]	En aquest o	cas està buit.
EXEMPLE	REQUEST: { "Identificad "Estat":"0", "Comentari "usuari":"at "password" } RESPONSE (Us {	or":"000000 ":"", oc1234", :"pwd" suari Trobat)	0000000000116",

, RESPONSE (Usuari no trobat) { "status": "ERROR", "dades": "Not found."	

API ENDPOINT	apiAddFitxa.php		
DESCRIPCIÓ	Permet modificar els camps de la fitxa d'un pacient.		
VERSIÓ	Usada per la versió Professional.		
MÈTODE	POST		
PARÀMETRES REQUEST	Format formulari HTML(form-data). També accepta format JSON.		
	Paràmetre	TIPUS {longitud }	Descripció
	ldentificado r	String {20}	NHC_Episodi únic per pacient. Format per 20 números
	Parentiu	String	Parentiu del contacte amb el pacient
	Apelatiu	String	Apel·latiu del pacient
	Contactes	String	Contacte del pacient
	Notes	String	Observacions del pacient
	Lliure	String	Observacions extra del pacient
	usuari	String	Paràmetre definit per validar la connexió a la api
	password	String	Paràmetre definit per validar la connexió a la api

PARÀMETRES RESPONSE	Objecte JSON amb la següent estructura:		
	Paràmetre	Descripció	
	status	"OK" = S'ha afegit correctament l'estat "Error" = No afegit correctament l'estat	
	dades[]	En aquest cas està buit.	
EXEMPLE	REQUEST:		
	<pre>{ "Identificador":"0000000000000000116", "Parentiu":"Fill", "Apelatiu":"Pepito", "Contactes":"Pau tlf: 7869423422", "Lliure":"", "usuari":"abc1234", "password":"pwd" }</pre>		
	RESPONSE (Usuari Trobat) {		
	"status": "OK",		
	"dades": []		
	}		
	RESPONSE (Usuari no trobat) { "status": "ERROR", "dades": "Not found." }		

4. DISSENY I IMPLEMENTACIÓ DE CLIENT FAMILIAR

4.1. ANDROID

4.1.1. Screenflow



4.1.2. Details de les pantalles

Pantalla	Descripció
<section-header><section-header><section-header><section-header></section-header></section-header></section-header></section-header>	Pantalla splash on trobem el nom de l'aplicació, els logos de l'Institut Català de la Salut i de la Generalitat de Catalunya, la versió de l'aplicació i finalment l'enllaç a canalsalut.gencat.cat
CAT ESP ENG Benvingut! Connecta COVID-19 et permet tenir informació sobre el teu familiar ingressat. Rebràs actualitzacions sobre el seu estat a través d'aquesta aplicació. També hi podràs trobar informació d'interès del centre on està ingressat i contactes de suport per a la gestió emocional d'aquesta situació. L'ús d'aquest servei suposa l'acceptació dels termes i la política de privacitat descrits en "Veure les condicions i polítiques." Veure les condicions i polítiques	Des d'aquesta pantalla l'usuari pot canviar l'idioma a català, castellà o anglès. També es troba amb un text que explica les funcionalitats de l'aplicació. Fent clic a Veure les condicions i polítiques accedirà a la pàgina web on les trobarà. Per poder accedir a l'aplicació, l'usuari haurà d'acceptar les condicions.



Pacients registrats
Cat

Nom:
Nom / Alies (opcional)

Ch (número targeta CatSalut)

TASA103010102

Codi identificador

D1234567890123456789

Codentificador

D1234567890123456789

Codence targeta CatSalut)

Primer cognom

COMENÇA EL SEGUIMENT

Comença en la comentaria de la comentaria de

Un cop s'han acceptat les condicions, ens trobem amb la següent pantalla. Fent clic al botó de la cantonada superior dreta podrem tornar a canviar l'idioma. En aquesta pantalla veurem la llista de pacients registrats a l'aplicació. En un inici està buida, però fent clic al botó central de la barra inferior podrem afegir el primer pacient.

Pantalla per afegir pacients. Per tal de poder afegir-lo, serà necessari conèixer el seu CIP, el codi identificador que li proporcionarà l'hospital, i el seu primer cognom. De manera opcional es pot indicar el nom del pacient, així es podrà diferenciar de la resta dins l'aplicació. Si no s'introdueix, a l'aplicació sortirà el CIP com a nom del pacient. En cas que un pacient no disposi de CIP, amb el codi identificador i el primer cognom es podrà afegir igualment.



Un cop hem afegit un pacient, arribem a aquesta pantalla. A la barra superior veiem el nom del pacient, un botó per editar el nom i un altre per actualitzar la informació. A continuació podem veure l'estat actual del pacient, la seva localització i la data de l'última actualització. Finalment veiem l'historial del pacient.



Si dins la pantalla del pacient fem clic al botó per editar el seu nom arribem a aquesta pantalla. Aquí veiem el nom actual del pacient, i podem introduir un de nou per canviar-lo.



Per veure la llista de pacients, hem de fer clic al botó esquerre de la barra inferior. Un cop hem afegit pacients, apareixeran a aquesta pantalla, i des d'aquí podrem veure un resum del seu estat actual, accedir a la seva informació completa (fent click en cadascun d'ells) o bé esborrar-los.



D'altra banda, si fem clic al botó dret de la barra inferior, arribarem a la pantalla d'enllaços d'interès. Des d'aquesta pantalla podrem accedir als enllaços de suport psicològic i d'informació sobre les visites als hospitals (aquest últim encara està pendent). Finalment, fent click a Sobre nosaltres, arribarem a l'última pantalla.



Pantalla amb informació sobre els desenvolupadors de l'aplicació.

4.1.3. Estructura i codi de l'aplicació

L'activitat principal, MainActivity, és on es troba la base de l'aplicació. La gestió de les altres classes passa gairebé sempre per ella, i és des de la qual controlem la interacció amb la llista de pacients, els enllaços d'interès i la modificació de paràmetres.

Per a gestionar la navegació entre pantalles de la barra inferior, hem creat un objecte BottomNavBar; detectem el botó que l'usuari apreta i canviem de fragment (que s'executa "a sobre" de l'activitat principal). Si es clica a afegir pacient, abans de passar a veure la llista obrim una finestra amb diferents entrades de text per a que l'usuari pugui omplir les dades del nou pacient. Es fa la comprovació del camp introduït, i si és incorrecte mostrem un missatge d'error en vermell.

Comptem amb 3 fragments diferents: HelpFragment, que gestiona l'apartat d'enllaços d'ajuda, PatientListFragment, que conté la llista de pacients i PatientFragment, que ens mostra la fitxa d'un sol pacient.

HelpFragment conté tres botons, dos que redirigeixen a les URL de suport emocional i horaris de visites definides en l'arxiu d'strings.xml; i un tercer, "sobre nosaltres", que crea una nova activitat (aboutUsActivity) que mostra una petita finestra amb informació sobre el projecte.

PatientListFragment conté la llista dels pacients guardats per l'usuari. S'utilitza un itemAdapter per a mostrar la informació de cada pacient en una llista d'elements, així com gestors d'alarma per a actualitzar periòdicament l'estat dels pacients.

En fer clic al botó d'afegir pacient, s'obtenen i es comproven les dades introduïdes, i en el cas de que aquest no estigui ja inclòs, s'afegeix a la llista. Per a transmetre les dades i fer la consulta a la base de dades, s'introdueixen en un objecte JSON que serà enviat al servidor amb la base de dades per a realitzar la consulta.

PatientFragment conté la fitxa detallada dels estats d'un pacient. S'obre en clicar en un dels elements de la llista de pacients. Per a mostrar l'historial d'estats es recuperen només per al pacient seleccionat, i es mostren amb l'ajuda de l'HistoryAdapter que té un disseny diferent.

Hi ha un botó que permet refrescar els estats, i un per a canviar el sobrenom introduït al pacient.

4.1.4. Crides remotes

La construcció i enviament de les peticions (POST) remotes es fan a partir de la classe Rider.java (classe asíncrona). Concretament la petició (en format JSON) es crea dins el mètode doInBackground() que és executat automàticament just després d'instanciar les extensions d'aquesta classe (presents a les classes AlarmReceiver.java i PatientListFragment.java que aporten les dades a enviar com a paràmetres del constructor) i cridar el mètode execute().

La connexió, enviament de la petició i l'obtenció de la resposta, en canvi, s'executen dins del mètode sendJsonReceiveJson() (que es crida al final de doInBackground()). La resposta d'aquest últim mètode es passa com a paràmetre als mètodes onPostExecute() de les extensions de Rider.java anteriorment mencionades (on es

27

tracten les respostes). Si la connexió NO és exitosa o el cos de la resposta no té un format JSON adequat es retornarà "null".

4.1.5. Tractament de dades rebudes

Els JSON's rebuts són passats com a paràmetre en els mètodes onPostExecute() de les classes AlarmReceiver.java i PatientListFragment.java. Allà s'analitzen els JSON's i si són correctes es creen (si no existien) i es desen a persistència els pacients amb les dades rebudes (mitjançant diferents mètodes de la classe Parser.java); l'actualització dels pacients es fa mitjançant l'enviament d'intents a la classe MainActivity.java que es qui s'encarrega de generar altra vegada uns nous intents específics per actualitzar cada classe pertinent i actualitzar les vistes.

En el cas de la classe AlarmReceiver.java (i si l'aplicació està oberta i visible) a més es genera la notificació pertinent.

4.1.6. Recursos de l'App

Tots els recursos gràfics de l'aplicació es poden trobar a la carpeta /src/main/res. En ella hi trobem diferents subcarpetes que contenen els diferents tipus de recursos utilitzats.

Les icones i imatges es troben a l'apartat drawable. Aquests s'utilitzen en les diferents classes, tant en els arxius .java com en els fitxers .xml d'aquestes.

Aquests fitxer .xml es troben a la subcarpeta de layout. Cada activitat i fragment té el seu fitxer gràfic associat, on hi apareix el disseny de pantalla amb la configuració de text, mides i colors dels elements que la conformen (botons, requadres de text, icones, barres de navegació, llistes…).

A la subcarpeta de values hi trobem els fitxers que contenen tota la informació "harcodejada" de l'aplicació, tant la definició de colors, com les dimensions, els textos i l'estil dels temes. Cadascun d'ells té el seu fitxer associat. El més important és el d'strings.xml, on es descriuen totes les paraules i frases que apareixen al llarg de les diverses pantalles. Hi ha dues subcarpetes més que contenen els mateixos textos en castellà i anglès per a fer possible el canvi d'idioma de l'app.

La gestió d'intercanvi d'idioma i de traducció es duu a terme mitjançant objectes de SharedPreferences (per guardar variables locals d'idioma) i amb un spinner que permet l'intercanvi un cop dins l'aplicació. La seva implementació es troba a la classe principal i al fragment de la llista de pacients.

En els fitxers de text hi trobem la definició d'uns estats fixos amb el seu número corresponent, tal com es va acordar. El número s'utilitzarà com a clau per a identificar l'estat que ens retorna el servidor, així no cal realitzar passos addicionals innecessaris a l'hora d'imprimir-los per pantalla.

4.1.7. Aspectes de seguretat

Tot i que la part més crítica i important és la implementació que s'ha fet de l'API en l'entorn servidor, nosaltres hem afegit algunes característiques per a limitar i fer més robustes les connexions al servidor remot.

- Per a limitar les vegades que un usuari de l'aplicació pot fer peticions al servidor (i per tant limitar les peticions que arribaran al servidor), s'ha implementat un mecanisme de Token Bucket (veure classe TokenBucket.java). Dues instàncies d'aquesta classe limiten quina quantitat de vegades per unitat de temps l'usuari pot refrescar un pacient o afegir (o intentar afegir) pacients nous.
- Els textos dels camps que l'usuari introdueix passen una verificació de format.
- Per a que la connexió https funcioni correctament s'ha afegit el certificat SSL del servidor (veure clau "PUBLIC_SERVER_CERT" del fitxer strings.xml). Aquest certificat s'hauria d'actualitzar en cas de caducitat.

29

4.2. IOS

4.2.1. Screenflow



4.2.2. Details de les pantalles

Pantalla	Descripció
Pantalla	 Descripció Aquesta és la pantalla on pots veure el llistat de pacients que tens i un resum del seu últim estat actualitzat. En prémer aquest botó: Demanes actualitzar tots els pacients de la llista. Pots tornar a canviar d'idioma dins l'App. Pots editar l'àlies que l'hi havies posat. Pots eliminar el pacient de l'App. accedeixes a la vista del pacient en detall, on
	s'hi pot veure el historial d'estats.
	6. Afegir un nou pacient.
Pacients Afegir Ajuda	7. Veure la pantalla d'ajuda.



WebView	PatientsView	AddView
5:23 중 ☞ Google A View this page in: English ▼ Translate	5:23 Pacients registrats C	5:23 र ॎ Afegir nou seguiment
SEM - Sistema d'Emergències Mèdiques Disclaimer Nota relativa a la posada en servei de noves solucions per a la prevenció i el control de la infecció pel SARS-CoV-2 Solució COVID19xat COVID19xat gestiona informació dels usuaris en relació amb l'afectació del COVID19 per determinar l'evolució de la malaltia i planificar les actuacions i respostes per part del situm de Salut.	Afegeix el teu primer pacient	Nom que li vulguis posar * Alias CIP AAAA0123456789 Codi identificador * 01234567890123456789
En aquest cas, la informació ha d'estar disponible abans que l'usuari accedeixi al COVID19xat i la informació també ha de ser fàcilment accessible a l'usuari una vegada ha accedit a l'aplicació. Missatge El COVID19xat facilita informació orientativa relativa a diferents aspectes de salut elaborada per professionals, i ofereix recomanacions, però en cap cas s'ha d'utilitzar per realitzar diagnòstics. El cont que proporciona no substitueix l'atenció personalitzada dels professionals de la salut.	□ o i Pacients Afegir Ajuda	Guardar = oS i Pacients Afegir Ajuda
	Un cop s'han acceptat les condicions, ens trobem amb la següent pantalla. Fent clic al botó de la cantonada superior dreta podrem tornar a canviar l'idioma. En aquesta pantalla veurem la llista de pacients registrats a l'aplicació. En un inici està buida, però fent clic al botó central de la barra inferior podrem afegir el primer pacient.	Pantalla per afegir pacients. Per tal de poder afegir-lo, serà necessari conèixer el seu CIP, el codi identificador que li proporcionarà l'hospital. S'indicarà el nom del pacient, per poder-lo diferenciar de la resta dins l'aplicació. En cas que un pacient no disposi de CIP, amb el codi identificador i el primer cognom es podrà afegir igualment.

AddView	AddSurnameView	Loading
5:25 ❤ ⊮ Afegir nou seguiment	5:25 ≎ ি Afegir nou seguiment	5:27 Pacient 1 /ID: POCA1710301003
Nom que li vulguis posar * Pacient 1 CIP Format incorrecte awd Codi identificador *	Nom que li vulguis posar * X Per verificar que coneixes aquest pacient, quin és el seu primer cognom? Cognom	 i Estat Previsió d'alta en les properes hores i Localització CENTRE_01 i Úttima actualització 2020-05-10 11:03:31
12345000000000000000000000000000000000000	Començar seguiment	Ingressat/da a planta CENTRE_01 2020-05-10 11:02:54 Trasllat de la UCI per anar a planta CENTRE_01 2020-05-10 10:53:49 Previsió d'alta en les properes hores
Image: space of the space of t	च o ^c i Pacients Afegir Ajuda	CENTRE_01
Serà necessari passar tots els controls de format on els errors s'indiquen amb vermell i surten les alertes pertinents.	Un cop passat aquest control, per fer una segona verificació del pacient s'haurà d'introduir el seu Cognom.	En haver afegit un pacient, ens trobem amb aquesta pantalla. On a la barra superior veiem el nom del pacient, el seu CIP o ID en cas de no disposar d'aquest primer i un botó per actualitzar la informació. A continuació podem veure l'estat actual del pacient, la seva localització i la data de l'última actualització. I en la part inferior trobem tot l'historial anterior en una llista desplaçable.

	LoadingView		DetailView		Alerta esc	combraria
5:28 Pacient /ID: POCA17	1 10301003 ℃	5:28 Pacient /ID: POCA17	€ € ک	6:12 Pacier	nts registra	
(i) Estat Efectes p I Loca CENTRE O Últin 2020-05-	Carregant • - • Carregant • - • Carreg	(i) Estat Efectes p I Loca CENTRE (2) Últin 2020-05-	eresonals entregats alització _01 na actualització 11 17:27:50 Previsió d'alta en les properes hores CENTRE_01 2020-05-10 11:03:31 Ingressat/da a planta		Pacient 1 Efectes per entregats 2020-05-11	1 sonals / û 17:27:50
	Trasllat de la UCI per anar a planta CENTRE_01	Ħ	CENTRE_01 2020-05-10 11:02:54	Estàs És po	s egur que vols e ossible que perdis realitzar aq	liminar aquest pacient? el codi de seguiment en uesta acció.
	2020-05-10 10:53:49 Previsió d'alta en les properes hores		Trasilat de la UCI per anar a planta CENTRE_01		Elim	linar
⊟ Pacients	o ^c s i Afegir Ajuda ────	= Pacients	oS i Afegir Ajuda		Canc	el·lar
				En fe l'escom aquesta	er clic braria a alerta de	al botó de ens apareix e seguretat.



En fer clic al botó d'editar ens apareix aquest PopUp on podem introduir un nou àlies per al pacient.

Per veure la llista de pacients, hem de fer clic al botó esquerre de la barra inferior. Un cop hem afegit pacients, els trobarem llistats en aquesta pantalla, i des d'aquí podrem veure un resum del seu estat actual, accedir a la informació seva completa (fent clic en cadascun d'ells) o bé esborrar-los.

D'altra banda, si fem clic al botó de barra inferior, dret la arribarem a la pantalla d'enllaços d'interès. Des d'aquesta pantalla podrem accedir als enllaços de suport psicològic i d'informació sobre les visites als hospitals últim encara està (aquest pendent). Finalment, fent click a Sobre nosaltres, arribarem a l'última pantalla.

aboutUsView		No internet
<section-header><image/><section-header><section-header><section-header><section-header><section-header><section-header><text><text></text></text></section-header></section-header></section-header></section-header></section-header></section-header></section-header>	AAAAO123456789 Error Has d'omplir tots els camps necessaris. Tancar AAAAO123456769 Error No tens connexió a internet, prova-ho més tard. Tancar Cr Error de credencials en afegir pacient. Tancar	
		च <mark>o</mark> S i Pacients Afegir Ajuda
Pantalla amb informació sobre els desenvolupadors de l'aplicació.	Alertes que es mostren durant la comprovació de credencials i l'actualització d'estats	Imatge que es mostra en no tenir connexió a internet i voler entrar a DetailView d'un pacient que no acabem d'actualitzar.

4.2.3. Estructura i codi de l'aplicació

L'estructura de la programació de pantalles de l'App és la següent:

Primer tenim ContentView que és on es controla quina és la vista que es mostra. Per fer-ho utilitzem un EnviornmentObject que ens permet tenir una variable comuna per saber quina vista és la que s'està mostrant en cada cas, li diem ViewRouter. Hi han 4 posibles vistes que són: ConditionsView, MainView, LanguageSelectView i WebView. Si ens trobem a MainView, hi tenim un TabView que ens permet navegar per les tres pàgines principals: PatientsView, AddView i InfoView.

PatientsView està estructurada de tal manera que amb una variable booleana podem distingir si mostrar la vista DetailView del pacient o la vista de la llista de pacients. Aquesta vista té un botó que permet mostrar la vista LanguageSelectView un altre cop. Aquí tenim també una subvista en format PopUp que permet canviar el nom del pacient, la vista ChangeNicknameView.

AddView és una vista senzilla amb TextFields i un botó. En prémer el botó, si els camps introduïts són correctes es solapa la vista AddSurnameView formant un PopUp.

InfoView només mostra tres botons, dos que redirigeixen a l'App del navegador del mòbil per obrir una URL externa i un botó que activa un PopUp que es solapa de la mateixa forma que l'anterior amb el nom de aboutUsView.

Així doncs, programat d'aquesta forma, on succeeixen la major part de funcionalitats de l'App és a la vista PatientsView.

Per simplificar el codi en totes aquestes vistes tenim les vistes més específiques en una classe a part anomenada subViews.

38

4.2.4. Elements de seguretat

Per tal d'assegurar que els inputs que es fan en afegir un pacient no comportin una injecció de codi en el servidor hem fet ús d'expressions regulars per comprovar que el format dels inputs és correcte. Així doncs també evitem fer peticions al servidor que no siguin vàlides. (Un exemple és el CIP, que sabem que té una estructura de 4 lletres majúscules i 10 números.)

4.2.5. Dades persistents de l'aplicació

Per emmagatzemar les dades persistents en l'App, utilitzem CoreData. Per fer-ho, definim una estructura de dades amb la informació bàsica de cada pacient necessària per realitzar les següents crides. Així com també l'informació més recent d'estat del pacient. Això permet que els pacients es mantinguin dins la llista i que l'usuari no hagi de tornar a introduir les seves credencials.

4.2.6. Crides remotes

Totes les crides remotes a l'API del servidor es realitzen utilitzant la funció Connectivity que es troba dins la classe RequestClass.

Aquesta funció verifica si l'estat de la connexió ha succeït correctament i ens retorna dues variables, una on s'indica l'estat d'aquesta connexió i una amb la informació en format JSON que retorna el servidor.

4.2.7. Tractament de dades rebudes

Com hem comentat, la funció anterior s'encarrega de tractar l'estat per tal de simplificar el codi en les classes View. Les dades retornades per la funció Connectivity en format JSON es deserialitzen utilitzant les estructures de la classe PacientInfo en cas de que aquesta no retorni cap dels possibles errors. Així doncs, si hi hagués un canvi en l'estructura d'aquest JSON caldria modificar aquestes estructures.

4.2.8. Recursos de l'App

 "Localizable.strings": En aquest arxiu se situen tots els textos fixos que poden aparèixer en algun moment a l'aplicació. No és un recurs com a tal, però cal tenir-ho en compte ja que són textos "hardcodejats".

Hi ha 3 arxius diferents d'aquest tipus que corresponen a cada idioma seleccionable a l'app i funcionen amb una estructura "key/value" de manera que totes les traduccions d'un mateix text estan associades a una key que serà el valor que utilitzem dins el software.

Un cop configurades les traduccions en aquests 3 arxius, el programa ja s'encarrega de fer les traduccions segons l'idioma actual de l'aplicació.

*En el cas dels estats, com vam acordar uns estats fixos que tenen una relació número/estat, vam decidir posar-los també en aquests fitxers d'idioma. En aquests fitxers hi definim la key com el número que ens retorna el servidor, així en imprimir-los per pantalla no cal fer passos innecessaris.

 Assets: en aquesta carpeta hi trobem totes les imatges i colors que s'utilitzen a l'app.

Entre les imatges hi trobem les icones associades a cada estat que podem mostrar, per fer aquesta associació hem creat una funció a la que passant-li el valor d'estat que ve del servidor, ens retorna el text del propi estat i el nom de la imatges que hem d'utilitzar en format String.

```
func StatusIcon(statusNum: String) -> (icon: String, status: String){
  let status = NSLocalizedString(statusNum, comment: "")
  var icon = ""
  switch statusNum {
    case "0": // "Pendent de triatge"
    icon = "arrow-decision"
    case "1": //
    icon = "hospital-box-outline"
    case "2": //
    icon = "hospital-building"
    case "3": //
```

}

```
icon = "clipboard-text"
case "4": //
  icon = "heart-pulse"
case "5": //
  icon = "account-arrow-right"
case "6": //
  icon = "phone-incoming"
case "7": //
  icon = "doctor"
case "8": //
  icon = "clock-check"
case "9": //
  icon = "ambulance"
case "10": //
  icon = "clipboard-check-outline"
case "11": //
  icon = "message-text"
default: //
  icon = "message-video"
}
return (icon, status)
```

4.3. WEB

En aquest apartat definirem tot l'entorn WEB involucrat en la versió Familiar de l'aplicatiu Connecta Covid 19. Començarem per la part més visual i exterior, i seguidament anirem desgranant la WEB fins al seu codi.

Tota la web familiar estará ubicada en:

https://wsm.ics.gencat.cat/connectaCovid/app/familiar/XXXX

4.3.1. Details de Pantalles

A diferència de les apps per dispositius mòbils, el flux de pantalles és més lineal, és per això que definim les pantalles en ordre de visualització una vegada s'ha accedit a la WEB familiar. El flux és:



Index.html: https://wsm.ics.gencat.cat/connectaCovid/app/familiar/

S/Sistema de Salut de Catalunya
 Benvingut! Aquesta aplicació et permetrà fer un seguiment personalitzat de l'status d'un o més familiars o pacients que hagin status infectats o tinguin símptomes del virus COVID-19 i estiguin sent atesos en algun centre sanitari de Catalunya. L'ús d'aquest servei suposa l'acceptació dels termes i la política de privacitat descrits en 'Veure les conditions i polítiques'. Per a continuar necessites acceptar aquestes condicions, pots consultarles fent click a l'enllaç de sota. He llegit i accepto les condicions Veure les condicions i polítiques
Accepto

Aquesta és la primera pantalla d'accés a la part Familiar. En aquesta es pot veure el disclaimer i hi ha un botó amb el qual es poden veure les condicions i polítiques de privacitat. Per accedir a la següent pantalla és necessari marcar la casella de 'He llegit i accepto les condicions'. Si no apareix un missatge d'error.

Veure le	s condicions i polítiques
	Accepto
Nol	has acceptat les condicions

autenticació.php:

https://wsm.ics.gencat.cat/connectaCovid/app/familiar/autenticacio.php



Una vegada acceptades les condicions i polítiques accedim a la pantalla d'introducció de les credencials del pacient del qual volem veure els seus estats. Per poder accedir als estats s'han de complir dues condicions:

 Omplir com a mínim els camps de Codi de Seguiment i Primer Cognom. Hi ha el cas on el pacient pot no tenir CIP.

Es obligatori insertar un codi de seguiment i els cognoms.

- 2. El format de les dades introduïdes ha de ser el correcte.
 - a. Codi Seguiment(Identificador): Aquest codi és l'anomenat Identificador format per NHC+Episodi i el seu format és de 20 números.
 - **b. CIP:** Format de 4 lletres majúscules a l'inici seguides de 6 números.
 - **c. Primer Cognom:** Ja que els cognoms poden contenir caràcters especials, l'única restricció es que el camp no pot contenir el carácter ';'.

Credencials incorrectes

Tot i superar les condicions, pot ser que les credencials introduïdes no es trobin a la DB, sigui per què són incorrectes o per què el pacient encara no ha sigut introduït a la DB de Connecta Covid 19. De la gestió d'aquest error s'encarrega la següent pantalla(Estat Pacient), nosaltres només veuríem un error en la pantalla d'autenticació.

No s'ha trobat el pacient, comprovi les credencials

estatpacient.php:

https://wsm.ics.gencat.cat/connectaCovid/app/familiar/estatPacient.php



Si les credencials introduïdes són correctes , veurem un llistat dels estats del pacient amb l'estat més actual el primer de la llista. En el cas que hi hagin més de 4 estats apareixerà en la part inferior esquerra de la llista un botó 'Més Estats' que permet veure més estats antics. També en la part inferior de la llista hi han dos botons, un per tornar enrere a la pantalla d'autenticació i un altre per actualitzar la pàgina.

A la part superior hi han dos botons, un que conté l'enllaç a informació i als horaris de visites dels hospitals, i l'altre a la web de gestió de les emocions.



Com a curiositat a la part inferior dreta de la pantalla hi ha un botó que desplega un pop-up informatiu sobre els desenvolupadors de Connecta Covid.



4.3.2. Estructura i codi de l'aplicació

A diferencia de l'APP Mòbil, la part WEB no té memòria d'estat ni de cap mena. És per això que l'estructura del codi és més "senzilla". Cada pantalla de la WEB familiar té el seu propi codi PHP/HTML amb el qual s'executa la WEB. Veurem que la majoria del codi és PHP ,HTML ,JavaScript les seves tecnologies/eines pròpies.

En aquest document no tindrem en compte el codi de les API, ja que només ens interessa el seu funcionament, ni tampoc el possible codi que estigui en execució en el servidor. Per tant tampoc entrarem en com està configurat el servidor.

Seguidament repassarem el codi de les pantalles en ordre d'aparició:

index.html:

Aquest codi és el més senzill de tota la part WEB. Ja que la pàgina només consta d'un disclaimer i d'una checkbox que s'ha de marcar per poder accedir a la següent pantalla. La única comprovació que hem de realitzar és mirar si l'usuari ha acceptat les condicions. Això s'aconsegueix mitjançant una funció de JavaScript anomenada 'validarCheckBox()', que comprova si s'ha marcat la casella. Si està marcada es procedeix a redirigir l'usuari a autenticació.php, en el cas contrari mostra un error que indica que s'han d'acceptar les condicions.

autenticacio.php

En aquesta pantalla primerament es comprova si hi s'ha fet GET amb una variable anomenada 'error', en aquest cas és sabut que l'usuari ha intentat accedir a les dades d'un pacient i aquest no existeix, llavors mostrem un error.

Ara quan l'usuari entra les credencials i comença el seguiment, abans d'enviar les dades al servidor es fa la comprovació de que les credencials estiguin en un format correcte. Això s'aconsegueix amb les funcions programades en JavaScript anomenades validarCodi() i formatValidation(), les quals comproven que els camps obligatoris estiguin omplerts i segueixin el format necessari. En cas erroni mostra un error informatiu i en cas correcte passem aquestes dades mitjançant POST a estatPacient.php

Aquestes dades van amb els noms: Identificador, CIP i Cognoms.

estatPacient.php

Per tal d'agafar les dades de la DB utilitzem la API apiFa.php, on li passem els paràmetres ja definits. Mirem si el status no és 'OK' (o si ens retorna un pacient sense estats), llavors vol dir que no hi ha cap pacient amb les credencials introduïdes i tornem a autenticacio.php amb un paràmatre GET de 'error=true'.

Per mostrar la taula d'estats volem que el primer estat estigui ressaltat, i que inicialment només es veigin 4 estats (incloent l'inicial), per mostrar la resta d'estats hi ha un botó (el qual executa la nostra funció changeView()). Per aconseguir la taula hi ha un script JavaScript el qual itera per tots els estats i diferencia el primer, els tres seguents i la resta, i els col·loca respectivament.

Estructura de crides:

Reutilitzant l'esquema anterior es veu com es passa l'informació per les diferents pàgines



4.3.3. Recursos

Per la confecció de la WEB em emprat diferents eines i tecnologies disponibles pel desenvolupament WEB.

El principal contingut del codi és:

- → HTML: HTML és el llenguatge estàndard utilitzat per definir la estructura de les pàgines web. Consisteix en una serie d'elements, atributs i 'tags' que indiquen al browser web com representar la pàgina web.
- → CSS(Cascading Style Sheets): És un llenguatge que descriu l'estil amb el que s'han de representar els elements HTML. HTML estableix l'estructura, CSS dona forma i color. CSS estalvia molt de temps ja pot controlar l'estil de varies pàgines amb un sol arxiu .css.
- → JavaScrip(JS): És un llenguatge de programació basat en objectes i de guió (script) que suporta els principis de polimorfisme, herència per delegació, abstracció i encapsulació. Està orientat a esdeveniments (principalment solen implementar-se als botons del codi HTML), se sol utilitzar per a generar codi HTML i és un llenguatge de programació interpretat.
- → PHP: PHP és un llenguatge de script de servidor i una potent eina per fer pàgines web dinàmiques i interactives. S'executa per la part de servidor i el client només visualitza la part HTML resultant de l'execució del script php. Un fitxer PHP pot contenir codi HTML, CSS i JavaScript. Utilitzem PHP per detectar les sol·licituds als arxius del servidor i fer consultes i gestions a la base de dades.

Com a recurs més visual tenim el directori on es troben els logos de CC19, del Sistema de Salut de Catalunya i el logo de cada estat convenientment modificat(p.e. Estat 3 => 3.svg) per poder mostrar un logo personalitzat per cada estat.

També importem la llibreria d'icones Font Awesome que permet inserir icones més intuïtives als botons.

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

4.3.4. Seguretat

Definim tres àrees de seguretat, en les pantalles, en el codi i en la connexió.

Des de les pantalles, només es pot accedir al contingut de la part familiar des de la pantalla de autenticació. Aquesta pantalla serà superada només si les credencials introduïdes són correctes i per això l'Identificador, CIP i el primer cognom són dades personals que només el familiar hauria de tenir. I en fer l'autenticació amb tres claus és difícil que algú aleatòriament ho encerti.

Des del codi, si s'intenta accedir a qualsevol pàgina de l'àmbit familiar través del seu URL, evitant així el login, automàticament es redirigeix a la pantalla de login.

Per exemple si algu introueix <u>https://.../familiar/autenticacio.php</u> al navegador, serà redirigit a <u>https://.../familiar/index</u>

A part, com hem vist en el apartat '5.2 Estructura i Codi de l'aplicació' les variables de validadació de la coneió estan codificades per no mostrarles en text pla.

```
$usuari = base64_decode('YzE5YXBw');
$pwd_x = base64_decode('NFJhc2VyODQ1M3Fw');
```

I per acabar, lés conexions WEB realitzades al servidor <u>wsm.ics.gencat.cat</u> compten amb la protecció del protocol HTTPS que inclou el xifrat amb TLS AES 256, amb claus de 256 bits. Per tant la informació entre Client<->Servidor es transfereix a través d'un canal xifrat segur.

5. DISSENY I IMPLEMENTACIÓ DE LA PART PEL PERSONAL MÈDIC

En aquest apartat definirem tot l'entorn WEB involucrat en la versió Professional de l'aplicatiu Connecta Covid 19 desenvolupat per ser utilitzat pel personal mèdic d'un centre. Començarem per la part més visual i exterior, i seguidament anirem desgranant la WEB fins al seu codi. A diferencia de la part versió Familiar, la Professional esta desenvolupada exclusivament en WEB. Tota la web professional estará ubicada en:

https://wsm.ics.gencat.cat/connectaCovid/app/professional/XXXX

Val a dir que hi ha un repositori /m que conté l'historial de pacient amb una interfície visual més amigable per dispositius mòbils. Aquesta versió es troba en la URL: <u>https://wsm.ics.gencat.cat/connectaCovid/app/professional/m/XXXX</u>

5.1.1. Detalls de Pantalles

Com s'ha exposat per la part WEB Familiar, a diferència de les apps per dispositius mòbils, el flux de pantalles és més lineal, és per això que definim les pantalles en ordre de visualització una vegada s'ha accedit al login de WEB professional.

A part, la versió Professional inclou mes funcionalitats que mostrarem en el diagrama de flux de pantalles.

El flux de pantalles és: (Les línies verdes representen acción realitzables des de cada pàgina)



loginProfessional.php:

https://wsm.ics.gencat.cat/connectaCovid/app/professional/loginProfessional.php

S/Sistema de Salut de Catalunya
Introdueix les teves dades Nom d'usuari: Contrasenya: Accedir

El login és la primera pantalla que veuran els professionals. Cada centre té un usuari d'accés vinculat com s'ha explicat en l'apartat '3.1 Estructura de de BBDD'. Per accedir a la taula de pacients ingressats en el centre s'han d'introduir correctament les credencials contingudes prèviament en la base de dades PROF.

Si les credencials són incorrectes es mostrarà el següent missatge:

Introdueix les teves dades	
Nom d'usuari:	\geq
Accedir	
Credencials Incorrectes	

taulaProfessional.php:

https://wsm.ics.gencat.cat/connectaCovid/app/professional/taulaProfessional.php

S/Sistema de Salut de Catalunya	Taula	de Pacients de U	Jniversitari Arnau	de Vilar	10va Hospital
€ Enrere					Formulari
Mostra 10 • registres			Cerca Pac	ient:	
Identificador 🔶 CIP 🔶 Nom 🔶	Cognoms	Estat 🔶	Data 🗸	Ült Act(m) ↔	Accions 🔶
00************************************		Pendent de Triatge	2020-05-22 17:00:05	40	/ =
00		Pendent de Triatge	2020-05-22 17:00:05	40	/

Aquesta pantalla és la més important, ja que és la que més informació i funcionalitats conté. A la capçalera a part del logo de /Salut, veurem un missatge personalitzat per cada centres: 'Taula de pacients de ···'.

Seguidament hi han dos botons, **'Enrere'** que torna l'usuari al login, i **'Formulari'** que desplega el formulari que permet afegir un estat nou a un pacient.

Identificador:	Cognom 1:	Cognom 2:	Estat:Escull un estat
	🛩 Act	ualitzar 関 🔁 🔟	

En el **'Formulari'** tres camps de text no-editables que s'omplen amb les dades del pacient des del qual s'hagi clicat al botó d'acció **'Estat Nou'**. Al costat d'aquests camps

trobem el selector d'estats, on podem triar quin estat afegir al pacient. Els estats mostrats son versions reduïdes de l'estat.(p.e. Trucada en 120' -> Us trucarem en els propers 120').



Sota dels camps i el selector, tenim els botons del formulari.

- El verd('Actualitzar Estat'), tramita el nou estat i renova la taula.
- El vermell('Cancel·lar Actualització'), esborra les dades del formulari.
- El blau('Actualitzar Estats'), renova la taula.
- El groc('Tancar Formulari'), tanca el formulari.

Després del formulari tenim la **'Taula de Pacients'**. La taula té diferents funcionalitats per fer més fàcil el seu ús:

Mostra	10 •	registres
	10	
furne i	25	
Iden	50	r
	100	

	Mostrant	de	1	a	10	de	2.406	registres
--	----------	----	---	---	----	----	-------	-----------

Aquesta opció permet modificar el nombre de pacients que es veuen en una pàgina de la taula.

Anterior	1	2	3	4	5		241	Següent
----------	---	---	---	---	---	--	-----	---------

Aquesta opció permet navegar a través de les diferents pàgines de la taula.

Cerca Pacient:	
----------------	--

Permet buscar un pacient en concret, es pot introduir des de el seu CIP, Identificador o simplement els seus cognoms.

Cada fila de la 'Taula Pacients' és un pacient. I cada columna és una dada dels pacients:

- Identificador, NHC+Episodi definit en el '3.1 Estructura de la BBDD'
 - CIP, codi identificador personal.
 - Nom
 - Cognoms
 - Estat, últim estat del pacient.
 - Data, data del últim estat.
 - Ùlt Act(Última Actualització), temps transcorregut en minuts des de l'ultim estat. Si el temps >1 dia el valor de la casella es '+1 dia'.
 - Accions, aquesta columna conte els botons d'acció que permet accedir a les seguents pantalles.

En la columna accions hi trobem els botons 'Estat Nou'



'Estat Nou' mou la pantalla fins el **'Formulari'** un cop clicat omple el 'Formulari' amb les dades del pacient de la fila del botó clicat. Només necessiten seleccionar un estat per poder tramitar el formulari.



Si fem clic en 'Historial' canviarem al historial del pacient de la fila del botó clicat.

Com a curiositat al final de la 'Taula de Pacients' hi ha un botó que desplega un popup informatiu sobre els desenvolupadors de la app.



h2.php:

https://wsm.ics.gencat.cat/connectaCovid/app/professional/h2.php

S/Sistema de Salut de Catalunya		HISTORIAL DE PEPITO
Mostra 10 ▼ registres	ESTATS Cerca Pacient:	Nom Pepito

Aquesta pantalla ens mostra per una part l'historial de **'Estats'** del pacient i per l'altre la seva **'Fitxa'**. A la capçalera a part del logo de /Salut, veurem un missatge personalitzat per cada centres: 'Historial de +el nom del pacient'.

La taula de **'Estats'** funciona d'igual manera que la **'Taula de Pacients'** de la pantalla taulaProfessional. Amb les mateixes característiques.

La **'Fitxa'** conté camps de text modificables pel personal mèdic. Una vegada modificats fent clic al botó 'Actualitzar Fitxa' s'actualitza la fitxa del pacient.

Per la part professional per mòbils(/m) l'historial del pacient té una altre vista però funciona exactament igual. Aquesta vista té un selector entre **'Estats'** i **'Fitxa'** a la part superior. A part els camps de la fitxa es poden activar/desactivar per evitar clics accidentals.



	ESTATS		
Mostra 10 🔻 registres		Cerca Pacient:	
Centre 🔶	Estat 🔶		Data
Universitari Arnau de Vilanova Hospital	Pendent de Triatge	2020-05-22 18:00:05	
Mostrant de 1 a 1 de 1 registres			Anterior 1 Següent
	Enrere Actualitza		

Estats Fitxa



5.1.2. Estructura i codi de l'aplicació

A diferencia de l'APP Mòbil, la part WEB no té memòria d'estat ni de cap mena. És per això que l'estructura del codi és més "senzilla". Cada pantalla de la WEB familiar té el seu propi codi PHP/HTML amb el qual s'executa la WEB. Veurem que la majoria del codi és PHP ,HTML ,JavaScript les seves tecnologies/eines pròpies.

En aquest document no tindrem en compte el codi de les API, ja que només ens interessa el seu funcionament, ni tampoc el possible codi que estigui en execució en el servidor. Per tant tampoc entrarem en com està configurat el servidor.

Seguidament repassarem el codi de les parts més clau de la versió professional:

taulaProfessional:

A l'inici del codi, en la part php, rebem les credencials del loginProfessional i fem la crida a la API 'apiPacientsLloc'. Avaluem la resposta i si el camp 'status' de la resposta el OK, guardem el JSON Object de la resposta en una variable per al seu futur ús en els JavaScripts continguts en el HTML de la pàgina. I si no és Ok tornarem a la pàgina de login. Si el 'status' ha sigut OK, podrem pasar a formar la 'Taula de Pacients' amb el plugin de jQuery 'DataTables'.

Primer definim la taula, important les columnes on anirà la informació dels pacients, amb HTML:

Seguidament utilitzem 'Datatables' i omplim la taula:

```
var table
            = $('#taulaP').DataTable(
         {
              "language": {
                            url: 'cat.json'
                      },
              "data": json.dades,
              "columns" : [
                  { data : "Identificador"},
                   { data : "CIP"},
                  { data : "Nom"},
{ data : "cognom1",
                  "render":function(data, type, full, meta){
return full.cognom1 +" "+full.cognom2;}},
                   { data : "Estat"},
{ data : "Data"},
                   { data : "Ultim"},
                   {defaultContent :
                       "<button type='button' class='editar btn btn-primary btn-sm'><i |
                            aria-hidden='true'></i></button><button type='button' class=
                            fa-list' aria-hidden='true'></i></button>"}
              ],
               'order": [[ 5, "desc" ]]
          });
```

Definicions:

 Language: Recurs del qual s'obtindrà la traducció dels textos de la taula.(p.e.
 Search->Cercar). En el nostre cas obtenim la traducció d'un fitxer XML en format json. - **Data:** Recurs del qual s'obtindra la informació per omplir la taula. En el nostre cas el JSON array d'estats que ens retorna la API.

 - Columns: Quin contingut anirà en cada columna, cada {} representa una columna.
 En el nostre cas hi posem les dades dels pacients contingudes en DATA. I en la última columna inserim contingut estàtic que són els botons d'acció.

- **Order:** Ordenem el contingut de la columna 5 de manera descendent. Recordem que les columnes comencen en el 0.

Una vegada feta la taula, una altre funció important és la que permet obtenir les dades de la fila des de la qual s'ha clicat al botó d'acció 'Estat Nou'. La funció obtenir_editar_data omple els camps del 'Formulari'.

```
var obtenir_editar_data = function(tbody,table){
    $(tbody).on("click", "button.editar", function(){
    var data = table.row( $(this).parents("tr")).data();
    //console.log(data);
    var id = $("#Identificador").val(data.Identificador),
        ubi = $("#Lloc").val(data.Lloc);
        cip = $("#CIP").val(data.CIP);
        nom = $("#nom").val(data.cognom1);
        cn = $("#cnom1").val(data.cognom1);
        cn2 = $("#cnom2").val(data.cognom2);
        ref = $("#refr").val("0");
        document.getElementById("intro").style.display = "block";
        document.documentElement.scrollTop = 0;
    });
}
```

Amb el 'Formulari' omplert, si fem clic al botó verd tramitarem el nou estat amb la funció JavaScript tramitar() que comprova que els camps no estiguin buits abans de cridar a la API 'apiAddEstat'.

Aquí ja podem veure un petit incís en un recurs de seguretat. Les variables que validen la connexió, usuari i password, no estan en text pla, sino que és descodifiquen.

```
function tramitar(){
     var x u = atob('YzE5YXBw');
     var x_pwd = atob('NFJhc2VyODQ1M3Fw');
     if(document.forms["submitEstat"]["Identificador"].value != "" &&
          document.forms["submitEstat"]["Estat"].value != ""){
     $.post("https://wsm.ics.gencat.cat/connectaCovid/apiAddEstat.php",
               {
                    Identificador: document.forms["submitEstat"]["Identificador"].value,
                    CIP: document.forms["submitEstat"]["CIP"].value,
Nom: document.forms["submitEstat"]["nom"].value,
cognom1: document.forms["submitEstat"]["cnom1"].value,
cognom2: document.forms["submitEstat"]["cnom2"].value,
Estat: document.forms["submitEstat"]["Estat"].value,
                    Lloc: document.forms["submitEstat"]["Lloc"].value,
                    usuari: x u,
                    password: x pwd
               }.
               function(data, status){
                    alert("Estat afegit correctament");
               });
     document.forms["submitEstat"].reset();
     document.getElementById('intro').style.display = 'none';
     window.location.reload();
     }else{
          alert("Es obligatori intoduir un Pacient i un Estat");
          return false;
     return false;
```

Abans de canviar de pantalla, explicarem la funció que permet traduir el valor numèric dels estats al seu valor real. Tot i que aquesta funció s'usa en qualsevol pantalla que mostri els estats, l'expliquem aqui i la seva funcionalitat és la mateixa a tot arreu.

```
function estatString(s){
        var estat;
        switch(s){
           case "0":
               estat= "Pendent de Triatge";
           break;
           case "1":
               estat = "Atès/a a urgències";
           break;
           case "2":
               estat = "Ingressat/da a planta";
           break;
            case "3":
               estat = "Pendent de Proves";
           break;
            case "4":
               estat = "Ingressat/da a UCI";
            break;
            case "5":
               estat = "Trasllat de la UCI per anar a planta";
            break;
           case "6":
              estat = "Us trucarem en els propers 120'";
           break;
           case "7":
              estat = "El metge/ssa ha fet valoració, situació estable";
           break;
           case "91":
              estat = "Previsió d'alta en les properes hores";
           break;
           case "9":
              estat = "Pendent de trasllat a un altre centre";
           break;
           case "92":
           estat = "Efectes personals entregats";
           break;
```

```
case "11":
    estat = "Possibilitat d'enviar missatge personalitzat";
break;
case "12":
    estat = "Possibilitat de videoconferència avui";
break;
default:
    estat= "En en el sistema";
}
return estat;
}
```

historial:

Si des de taulaProfessional fem clic al botó de l'historial d'un pacient, anirem a la pantalla de historial, des de la qual amb les dades de pacient facilitades fem una crida a la API 'apiPacientFitxa', que ens retorna els estats i la fitxa del pacient.

Pel que fa al tractament dels estats, seguim el mateix procediment que per la 'Taula de Pacients'. Definim la taula amb HTML i amb 'DataTables' la omplim.

```
var table = $('#taulaP').DataTable(
    {
        "language": {
            url: 'cat.json'
        },
        "data": json.dades.Estats,
        "columns" : [
            { data : "Lloc"},
            { data : "Estat"},
            { data : "Data"},
        ],
        "order": [[ 2, "asc" ]]
    });
```

La novetat és la creació de la fitxa a partir de la resposta de la API. El que fem és similar a la taula, definim amb HTML els camps de les fitxes i amb jQuery els omplim.

```
$(document).ready(function(){
    var json = <?php echo $jsonObject; ?>;
    $("#parentiu").val(json.dades.Fitxa.Parentiu);
    $("#apl").val(json.dades.Fitxa.Apelatiu);
    $("#pC").val(json.dades.Fitxa.Contactes);
    $("#notes").val(json.dades.Fitxa.Notes);
    $("#cE").val(json.dades.Fitxa.Lliure);
```

});

Per actualitzar la fitxa de pacient, només cal fer clic en el botó 'Actualitza Fitxa' i es cridarà a la funció tramitarFitxa().

```
function tramitarFitxa(){
    var id = '<?php echo $id; ?>';
    var x u = atob('YzE5YXBw');
    var x_pwd = atob('NFJhc2VyODQ1M3Fw');
    $.post("https://wsm.ics.gencat.cat/connectaCovid/apiAddFitxa.php",
            {
                Identificador: id,
                Parentiu: document.forms["fitxa"]["parentiu"].value,
                Apelatiu: document.forms["fitxa"]["apl"].value,
                Contactes: document.forms["fitxa"]["pC"].value,
                Notes: document.forms["fitxa"]["notes"].value,
                Lliure: document.forms["fitxa"]["cE"].value,
                usuari: x_u,
                password: x_pwd
            function(data, status){
                alert("Fitxa Actualitzada Correctament");
            });
    return false;
```

5.1.3. Recursos

Per la confecció de la WEB em emprat diferents eines i tecnologies disponibles pel desenvolupament WEB.

El principal contingut del codi és:

- → HTML: HTML és el llenguatge estàndard utilitzat per definir la estructura de les pàgines web. Consisteix en una serie d'elements, atributs i 'tags' que indiquen al browser web com representar la pàgina web.
- → CSS(Cascading Style Sheets): És un llenguatge que descriu l'estil amb el que s'han de representar els elements HTML. HTML estableix l'estructura, CSS dona forma i color. CSS estalvia molt de temps ja pot controlar l'estil de varies pàgines amb un sol arxiu .css.
- → JavaScrip(JS): És un llenguatge de programació basat en objectes i de guió (script) que suporta els principis de polimorfisme, herència per delegació, abstracció i encapsulació. Està orientat a esdeveniments (principalment solen implementar-se als botons del codi HTML), se sol utilitzar per a generar codi HTML i és un llenguatge de programació interpretat.
- → PHP: PHP és un llenguatge de script de servidor i una potent eina per fer pàgines web dinàmiques i interactives. S'executa per la part de servidor i el client només visualitza la part HTML resultant de l'execució del script php. Un fitxer PHP pot contenir codi HTML, CSS i JavaScript. Utilitzem PHP per detectar les sol·licituds als arxius del servidor i fer consultes i gestions a la base de dades.

A part de les eines principals també hem fet servir:

- → jQuery: És una biblioteca o framework de Javascript, que permet simplificar la manera interaccionar amb els documents HTML, manipular l'arbre DOM, gestionar esdeveniments, desenvolupar animacions i afegir interacció amb la tecnologia AJAX en pàgines web. AJAX és una tècnica que permet accedir a servidors web des d'una web ja carregada. Permet enviar/rebre d'un servidor i actualitzar zones de la pàgina web sense necessitat d'actualitzar la pàgina sencera.jQuery ofereix una sèrie de funcionalitats basades en Javascript que d'una altra manera requerien molt més codi.
- → DataTables: És un plugin per jQuery que generar de forma ràpida i senzilla taules de dades amb tota mena de funcionalitats.

Tant jQuery com Datatables s'han d'importar en el codi.

```
<script src="https://code.jquery.com/jquery-3.3.1.js"></script>
<link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css"/>
```

Com a recurs més visual tenim el directori on es troben els logos de CC19, del Sistema de Salut de Catalunya i el logo de cada estat convenientment modificat(p.e. Estat 3 => 3.svg) per poder mostrar un logo personalitzat per cada estat.

També importem la llibreria d'icones Font Awesome que permet inserir icones més intuïtives als botons.

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"><//com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"></com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css</com/ajax/libs/font-awesome/4.7.0/css/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0/css/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/4.7.0</com/adatas/font-awesome/font-awesome/font-awesome/font-awesome/font-awesome/f

5.1.4. Seguretat

Definim tres àrees de seguretat, en les pantalles, en el codi i en la connexió.

Des de les pantalles, només es pot accedir al contingut de la part professional des de la pantalla de login. Aquesta pantalla serà superada només si les credencials introduïdes són correctes i per això l'usuari i la contrasenya són codis alfanumèric augmentant la seguretat prevenint que qualsevol persona els pugui deduir.

Des del codi, si s'intenta accedir a qualsevol pàgina de l'àmbit professional a través del seu URL, evitant així el login, automàticament es redirigeix a la pantalla de login.

Per exemple si algu introueix <u>https://.../professional/taulaProfessional.php</u> al navegador, serà redirigit a <u>https://.../professional/loginProfessional.php</u>.

A part, com hem vist en el apartat '5.2 Estructura i Codi de l'aplicació' les variables de validadació de la coneió estan codificades per no mostrarles en text pla.

```
$usuari = base64_decode('YzE5YXBw');
$pwd x = base64 decode('NFJhc2VyODQ1M3Fw');
```

I per acabar, lés conexions WEB realitzades al servidor <u>wsm.ics.gencat.cat</u> compten amb la protecció del protocol HTTPS que inclou el xifrat amb TLS AES 256, amb claus de 256 bits. Per tant la informació entre Client<->Servidor es transfereix a través d'un canal xifrat segur.