

Universitat de Girona
Escola Politècnica Superior

Grau en Enginyeria Informàtica

PROJECTE FINAL DE GRAU

**Eina de suport per a l'elaboració dels
horaris dels graus de l'EPS**

Autor:

Adrià Ribas Chico

Tutors:

Dra. Marta Fort Masdevall
Dr. Antonio Rodríguez Benítez

MEMÒRIA

Convocatòria:

Gener de 2023

Departament:

Informàtica, Matemàtica Aplicada i Estadística

Projecte: Projecte Final de Grau
Document: Memòria
Títol: Eina de suport per a l'elaboració dels
horaris dels graus de l'EPS
Autor: Adrià Ribas Chico
Data: 10 de gener de 2023

Estudi:
Grau en Enginyeria Informàtica
Universitat de Girona

Tutora 1:
Dra. Marta Fort Masdevall
Universitat de Girona
Informàtica, Matemàtica Aplicada i Estadística
marta.fort@udg.edu

Tutor 2:
Dr. Antonio Rodríguez Benítez
Universitat de Girona
Informàtica, Matemàtica Aplicada i Estadística
antonio.rodriguez@udg.edu

Agraïments

Per començar, vull reconèixer a la Marta Fort i a l'Antonio Rodríguez tots els consells, ajudes i, sobretot, el temps que m'han dedicat.

A més, vull donar les gràcies a la meva família pel suport incondicional i, en especial, a l'Alba per formar part d'aquest projecte.

Índex

1	Introducció	1
1.1	Antecedents	1
1.2	Propòsit	1
1.3	Motivacions	2
1.4	Objectius generals	3
1.5	Estructura del document	4
2	Estudi de viabilitat	6
2.1	Viabilitat tecnològica	6
2.2	Costos de personal	6
3	Metodologia	8
3.1	Gestió del projecte	8
3.2	Metodologia de desenvolupament	9
4	Marc de treball i conceptes previs	10
4.1	Nomenclatura específica	10
4.2	Funcionament actual	14
4.3	Document d'un pla docent	16
4.4	Conceptes tècnics	18
4.4.1	Introducció	18
4.4.2	Aplicació client	19
4.4.3	Aplicació servidor	20
4.4.4	Comunicació entre client i servidor	21
4.4.5	Emmagatzematge i gestió de dades	21
5	Requisits del sistema	23
5.1	Consideracions inicials	23
5.2	Requisits funcionals	24
5.2.1	Requisits generals	24
5.2.2	Requisits dels Administradors	25
5.2.3	Requisits dels Coordinadors	27
5.2.4	Requisits dels Directors de departament	29
5.2.5	Requisits dels Responsables de docència	30
5.2.6	Requisits dels Professors	30
5.3	Requisits no funcionals	31
5.4	Requisits de domini	31

5.5	Matriu de dependències	31
6	Planificació	33
6.1	Descomposició i planificació de tasques	33
6.1.1	Consideracions inicials	33
6.1.2	Visió general	35
6.1.3	Gestió del projecte	36
6.1.4	Anàlisi i disseny	38
6.1.5	Desenvolupament	40
6.1.6	Proves	49
6.1.7	Llançament a producció	50
6.2	Matriu de traçabilitat	52
6.3	Temporalització	53
6.3.1	Determinació d'activitats	53
6.3.2	Diagrama d'activitats	60
6.3.3	Cronograma	62
7	Estudis i decisions	63
7.1	Gestió i estructuració del projecte	63
7.1.1	Control de versions	63
7.1.2	Estructuració del projecte	64
7.2	Aplicació client	65
7.2.1	<i>Bundler</i>	65
7.2.2	<i>Framework</i>	66
7.2.3	Enrutament	68
7.2.4	Gestió d'estats	68
7.2.5	Estètica i funcionalitats addicionals	68
7.2.6	Proves	70
7.3	Aplicació servidor	70
7.3.1	Entorn i llenguatge de programació	70
7.3.2	<i>Framework</i>	71
7.3.3	ORM	71
7.3.4	Proves	72
7.4	Base de dades	72
7.4.1	Entorn d'execució	72
7.4.2	Sistema gestor de bases de dades	73
7.4.3	Proves	73
7.5	Eines de desenvolupament	74
7.5.1	Editor de codi	74
7.6	Confecció de la memòria	75
7.6.1	Elaboració del document	75

7.6.2	Disseny de les figures	75
7.6.3	Diagrama de Gantt	75
8	Anàlisi i disseny del sistema	77
8.1	Anàlisi dels casos d'ús	77
8.1.1	Actors implicats	77
8.1.2	Autenticació	78
8.1.3	Gestió del pla docent	79
8.1.4	Gestió d'usuaris	82
8.1.5	Visualització d'horaris	83
8.1.6	Modificació d'horaris d'estudis	84
8.2	Disseny de la base de dades	87
8.2.1	Model entitat-relació	87
8.2.2	Descripció de les entitats	88
8.3	Interfícies d'usuari	94
8.3.1	Interfícies d'autenticació	94
8.3.2	Interfícies de gestió del pla docent	97
8.3.3	Interfícies de gestió d'usuaris	98
8.3.4	Interfícies de consulta i elaboració d'horaris	101
9	Implementació i proves	106
9.1	Aplicació servidor	106
9.1.1	Configuració, variables d'entorn i <i>logging</i>	106
9.1.2	Seguretat i autenticació	108
9.1.3	Models de dades	110
9.1.4	Enviament de correus electrònics	112
9.1.5	Implementació de l'API	113
9.1.6	Proves de la base de dades	114
9.1.7	Proves de l'API	116
9.2	Aplicació client	120
9.2.1	Enrutament	120
9.2.2	Solapaments de tipus de laboratori	123
9.2.3	Usabilitat i estètica	126
9.2.4	Proves	131
10	Implantació i resultats	135
10.1	Implantació en l'entorn local	135
10.2	Resultats obtinguts	137
10.2.1	Consideracions inicials	137
10.2.2	Autenticació	138
10.2.3	Gestió del pla docent	143

10.2.4 Gestió d'usuaris	150
10.2.5 Consulta i elaboració d'horaris	153
11 Conclusions	170
11.1 Assoliment dels objectius i els requisits	170
11.2 Valoració personal	171
12 Treball futur	172
12.1 Ampliacions i millores bàsiques	172
12.2 Ampliacions i millores addicionals	173
Bibliografia	174

Índex de figures

4.1	Exemple d'una plantilla de PowerPoint utilitzada per a l'elaboració d'horaris.	15
4.2	Fragment del full d'un estudi d'un fitxer de pla docent.	17
4.3	Fragment del full dels tipus de laboratori d'un fitxer de pla docent.	18
4.4	Esquema de l'arquitectura d'una aplicació web convencional. . .	18
6.1	Descomposició general de les tasques del projecte.	35
6.2	Esquema de la branca de planificació <i>Gestió del projecte</i>	36
6.3	Esquema de la branca de planificació <i>Anàlisi i disseny</i>	38
6.4	Esquema de la branca de planificació <i>Desenvolupament</i>	40
6.5	Esquema de la branca de planificació derivada <i>Implementació de l'aplicació servidor</i>	41
6.6	Esquema de la branca de planificació derivada <i>Implementació del client</i>	45
6.7	Esquema de la branca de planificació <i>Proves</i>	49
6.8	Esquema de la branca de planificació <i>Llançament a producció</i> . . .	51
6.9	Diagrama de les activitats planificades.	60
6.10	Diagrama de Gantt.	62
7.1	Logotip de Git.	63
7.2	Logotip de GitHub.	64
7.3	Logotip de npm.	65
7.4	Logotip de Vite.	66
7.5	Logotip de Vue.	67
7.6	Logotip de Pinia.	68
7.7	Logotip de Quasar.	69
7.8	Logotip de Chrome DevTools.	70
7.9	Logotip de Node.js.	71
7.10	Logotip d'Express.js.	71
7.11	Logotip de Sequelize.	72
7.12	Logotip de Postman.	72
7.13	Logotip de Docker.	73
7.14	Logotip de MySQL.	73
7.15	Logotip de DataGrip.	74
7.16	Logotip de Visual Studio Code.	74
7.17	Logotip de LaTeX.	75
7.18	Logotip de Diagrams.net.	75
7.19	Logotip de Teamgantt.	76

8.1	Jerarquia d'actors que poden interactuar amb l'aplicació.	77
8.2	Diagrama dels casos d'ús relacionats amb l'autenticació.	78
8.3	Diagrama de casos d'ús general de la gestió del pla docent.	79
8.4	Diagrama de casos d'ús de la pujada d'un pla docent.	80
8.5	Diagrama de casos d'ús de la modificació del pla docent.	81
8.6	Diagrama de casos d'ús de la gestió d'usuaris.	82
8.7	Diagrama de casos d'ús de la visualització d'horaris.	83
8.8	Diagrama de casos d'ús general de la modificació d'horaris.	84
8.9	Diagrama de casos d'ús de la modificació d'un horari d'un estudi.	85
8.10	Diagrama de casos d'ús de la visualització de solapaments.	86
8.11	Diagrama relacional de la base de dades.	88
8.12	Disseny de la interfície d'autenticació.	95
8.13	Disseny de la interfície de restabliment de contrasenya.	95
8.14	Disseny de la interfície de confirmació d'enviament d'un <i>email</i> per al restabliment de la contrasenya.	96
8.15	Disseny de la interfície d'elecció d'una nova contrasenya.	96
8.16	Disseny de la interfície de la pujada d'un pla docent.	97
8.17	Disseny de la interfície de gestió del pla docent.	98
8.18	Disseny de la interfície d'assignació de Coordinadors a estudis.	99
8.19	Disseny de la interfície de control de Coordinadors.	99
8.20	Disseny de la interfície de creació de Coordinadors.	100
8.21	Disseny de la interfície d'accés a la gestió d'un horari d'un estudi.	101
8.22	Disseny de la interfície d'accés a la visualització d'un horari d'un Professor.	102
8.23	Disseny de la interfície d'accés a la visualització d'un horari d'u- na aula.	103
8.24	Disseny de la interfície de visualització d'un horari d'un estudi.	104
8.25	Disseny de la interfície de modificació d'un horari d'un estudi.	105
9.1	Fragment del fitxer de configuració aplicat a l'entorn de desen- volupament.	107
9.2	Definició del format de les credencials mitjançant Joi.	108
9.3	Comparació d'una contrasenya amb una contrasenya encriptada mitjançant bcrypt.	108
9.4	Encriptació d'una contrasenya mitjançant bcrypt.	109
9.5	Generació d'un JWT mitjançant jwt.	109
9.6	Validació d'un <i>token</i> d'autenticació mitjançant jwt.	110
9.7	Fragment de la definició del model de dades per als estudis.	111
9.8	Funció que defineix les relacions del model de dades per als es- tudis.	112
9.9	Taules de la base de dades generades per l'aplicació servidor.	115

9.10	Atributs de la taula <i>Studies</i>	115
9.11	Resposta del servidor quan una petició no duu el <i>token</i> per a l'autorització.	116
9.12	Resposta del servidor quan el <i>token</i> per a l'autorització d'una petició és invàlid.	117
9.13	Resposta del servidor quan un usuari intenta esborrar-ne un altre i no té els permisos suficients.	117
9.14	Exemple d'una petició de filtratge d'assignatures.	118
9.15	Fragment de la resposta de l'aplicació servidor davant d'una petició de filtratge d'assignatures.	119
9.16	Funció que s'executa abans d'entrar a qualsevol ruta.	120
9.17	Fragment de la definició de la ruta de visualització o modificació d'horaris d'estudis.	122
9.18	Exemple de la divisió d'un dia en franges per a calcular l'ocupació de les aules del tipus de laboratori "INFOR".	124
9.19	Petició per recuperar la informació necessària per calcular solapaments de tipus de laboratori.	125
9.20	Algoritme que divideix els blocs horaris d'un dia concret en grups i columnes.	127
9.21	Algoritme que calcula el nombre de columnes que pot ocupar horitzontalment un bloc horari.	128
9.22	Funció que retorna la funció que permet obtenir el color d'un element concret segons un tipus de grup.	129
9.23	Targeta que indica l'hora d'inici, de fi i duració exactes d'un bloc horari.	130
9.24	Blocs horaris marcats com a solapats.	130
9.25	Contingut del desplegable d'usuaris quan ja n'hi ha algun d'assignat.	131
9.26	Distribució de blocs horaris al calendari en una situació complexa.	132
9.27	Fragment d'un horari vist des de la vista setmanal general.	133
9.28	Fragment d'un horari vist des de la vista de setmanes A.	133
9.29	Fragment d'un horari vist des de la vista de setmanes B.	134
9.30	Detall dels solapaments d'un tipus de laboratori determinat.	134
10.1	Interfície resultant d'autenticació.	138
10.2	Interfície resultant de restabliment de contrasenya.	139
10.3	Interfície resultant de confirmació d'enviament d'un <i>email</i> per al restabliment de la contrasenya.	140
10.4	Interfície resultant d'elecció d'una nova contrasenya.	141
10.5	Interfície resultant de tancament de sessió.	142
10.6	Interfície resultant de la pujada d'un pla docent.	143

10.7	Interfície resultant de visualització i gestió dels estudis i assignatures d'un pla docent.	144
10.8	Interfície resultant de modificació d'una assignatura.	145
10.9	Interfície resultant de selecció d'una àrea per assignar-la a una assignatura.	146
10.10	Interfície resultant de visualització dels departaments i àrees d'un pla docent.	147
10.11	Interfície resultant de visualització i gestió dels tipus de laboratori d'un pla docent.	148
10.12	Interfície resultant de modificació d'un tipus de laboratori.	149
10.13	Interfície resultant d'assignació de Coordinadors a estudis.	150
10.14	Interfície resultant de la gestió de Coordinadors.	151
10.15	Interfície resultant de creació d'un nou Coordinador.	152
10.16	Interfície resultant d'accés a visualitzar o a modificar un horari d'un estudi.	153
10.17	Interfície resultant de visualització d'un horari d'un estudi.	155
10.18	Interfície resultant de la visualització d'un horari horari d'un estudi en la vista de setmanes B.	155
10.19	Interfície resultant de visualització de la informació d'un bloc horari.	157
10.20	Interfície resultant de visualització de la informació d'un bloc horari genèric.	158
10.21	Interfície resultant de modificació d'un horari d'un estudi.	160
10.22	Interfície resultant de modificació d'un bloc horari.	162
10.23	Interfície resultant de modificació d'un bloc horari genèric.	163
10.24	Interfície resultant de visualització de la informació detallada sobre els solapaments de tipus de laboratori.	164
10.25	Interfície resultant de visualització de possibles solapaments a l'horari.	165
10.26	Interfície resultant de visualització de marques en els blocs horaris que es solapen.	166
10.27	Interfície resultant de configuració de la modificació d'un horari.	167
10.28	Interfície resultant de gestió de blocs horaris.	168
10.29	Interfície resultant de la gestió de blocs horaris genèrics.	169

Índex de taules

2.1	Taula de costos de personal.	7
6.1	Taula del paquet de treball <i>Definició dels requisits</i>	36
6.2	Taula del paquet de treball <i>Planificació de tasques</i>	37
6.3	Taula del paquet de treball <i>Documentació</i>	37
6.4	Taula del paquet de treball <i>Disseny de l'arquitectura</i>	38
6.5	Taula del paquet de treball <i>Disseny de la base de dades</i>	39
6.6	Taula del paquet de treball <i>Disseny de les interfícies d'usuari</i>	39
6.7	Taula del paquet de treball <i>Preparació inicial de l'aplicació servidor</i>	42
6.8	Taula del paquet de treball <i>Configuració de la base de dades</i>	42
6.9	Taula del paquet de treball <i>Definició dels models de dades</i>	43
6.10	Taula del paquet de treball <i>Implementació de l'enrutament</i>	43
6.11	Taula del paquet de treball <i>Implementació del login i del registre</i>	44
6.12	Taula del paquet de treball <i>Implementació dels serveis CRUD</i>	44
6.13	Taula del paquet de treball <i>Preparació inicial de l'aplicació client</i>	45
6.14	Taula del paquet de treball <i>Implementació del login i relacionats</i>	46
6.15	Taula del paquet de treball <i>Implementació de la part dels Administradors</i>	46
6.16	Taula del paquet de treball <i>Implementació de la part dels Coordinadors</i>	47
6.17	Taula del paquet de treball <i>Implementació de la part dels Directors de departament</i>	47
6.18	Taula del paquet de treball <i>Implementació de la part dels Responsables de docència</i>	48
6.19	Taula del paquet de treball <i>Implementació de la part dels Professors</i>	48
6.20	Taula del paquet de treball <i>Testeig de l'aplicació servidor</i>	49
6.21	Taula del paquet de treball <i>Testeig de l'aplicació client</i>	50
6.22	Taula del paquet de treball <i>Testeig general de casos d'ús</i>	50
6.23	Taula del paquet de treball <i>Preparació de l'aplicatiu</i>	51
6.24	Taula del paquet de treball <i>Posada en marxa al servidor de hosting</i>	51
6.25	Taula de l'activitat <i>Definició de requisits</i>	53
6.26	Taula de l'activitat <i>Planificació de tasques</i>	54
6.27	Taula de l'activitat <i>Disseny de l'arquitectura</i>	54
6.28	Taula de l'activitat <i>Disseny de la base de dades</i>	54
6.29	Taula de l'activitat <i>Disseny de les interfícies d'usuari</i>	54
6.30	Taula de l'activitat <i>Preparació inicial de l'aplicació servidor</i>	55
6.31	Taula de l'activitat <i>Configuració de la base de dades</i>	55

Índex de taules

6.32	Taula de l'activitat <i>Definició dels models de dades.</i>	55
6.33	Taula de l'activitat <i>Implementació de l'enrutament.</i>	55
6.34	Taula de l'activitat <i>Preparació inicial de l'aplicació client.</i>	56
6.35	Taula de l'activitat <i>Implementació del login i relacionats.</i>	56
6.36	Taula de l'activitat <i>Implementació de la part dels Administradors.</i>	56
6.37	Taula de l'activitat <i>Implementació de la part dels Coordinadors.</i>	56
6.38	Taula de l'activitat <i>Implementació de la part dels Directors de departament.</i>	57
6.39	Taula de l'activitat <i>Implementació de la part dels Responsables de docència.</i>	57
6.40	Taula de l'activitat <i>Implementació de la part dels Professors.</i>	57
6.41	Taula de l'activitat <i>Testeig de l'aplicació servidor.</i>	58
6.42	Taula de l'activitat <i>Testeig de l'aplicació client.</i>	58
6.43	Taula de l'activitat <i>Testeig general de casos d'ús.</i>	58
6.44	Taula de l'activitat <i>Posada en marxa de l'aplicatiu.</i>	58
6.45	Taula de l'activitat <i>Documentació.</i>	59

Introducció

En aquest capítol, es situarà el marc del projecte. Més concretament, se'n presentaran els antecedents (veure secció 1.1), el propòsit (veure secció 1.2), les motivacions (veure secció 1.3) i els objectius generals (veure secció 1.4). A més a més, es detallarà l'estructura del document (veure secció 1.5).

1.1 Antecedents

En aquesta secció, es resumiran els antecedents que han donat peu al plantejament d'aquest projecte.

La idea del projecte neix de determinades necessitats que el personal de l'Escola Politècnica Superior de la Universitat de Girona fa temps que té. Més concretament, es tracta d'una necessitat del personal encarregat de gestionar tot el que fa referència a la confecció i manteniment dels horaris del centre: horaris dels estudis, dels professors, ocupació d'aules i espais, etc.

Actualment, en els processos d'elaboració d'horaris i de distribució de docència entre professors hi intervenen més de quaranta persones, les quals utilitzen mètodes i eines relativament incòmodes i poc àgils, no automatitzades ni específicament dissenyades per abordar aquest tipus de tasques. Tampoc existeix cap plataforma que unifiqui les fases d'aquest procés de gestió ni que n'estableixi una manera de fer comuna. A més, no tenen manera de detectar incompatibilitats horàries ni solapaments *a priori* de forma automàtica. Per aquests motius, en moltes ocasions s'han de repetir certes fases del procés fins que el resultat és vàlid i la gent implicada hi està d'acord.

Al capítol 4 es descriurà amb més profunditat com funcionen avui dia aquests processos d'elaboració i gestió d'horaris de l'escola.

1.2 Propòsit

En aquesta secció, s'exposarà el propòsit general del projecte tenint en compte els antecedents vists a la secció 1.1.

En definitiva, la gestió dels horaris de l'EPS suposa una inversió de temps massa elevada per a la gent que se n'ocupa. És per això que Marta Fort Masdevall, coordinadora d'estudi del Grau en Enginyeria Informàtica de la universitat, proposa un projecte de fi de grau pensat per a dues persones que té la finalitat de trobar una solució al problema.

El propòsit és desenvolupar una eina de suport informàtic que permeti a l'usuari elaborar horaris de forma àgil, eficient i segura. L'eina també hauria de ser capaç de comprovar automàticament la disponibilitat de les aules, les incompatibilitats horàries dels professors i la concordança entre les assignatures i el nombre de grups previstos. A més a més, hauria d'oferir diverses vistes per tal que l'usuari pugui visualitzar la informació, com ara:

- Els horaris d'un estudi per curs i quadrimestre.
- Els horaris d'un professor per quadrimestre.
- L'ocupació d'un espai per quadrimestre.

També es planteja la possibilitat de disposar d'un sistema de control d'usuaris. Cada usuari tindria assignat un conjunt de rols determinat. Els rols representarien els diferents càrrecs del personal de l'EPS en matèria de gestió d'horaris. Així doncs, cada usuari podria executar les accions i consultar la informació que el seu conjunt de rols li permeti. D'aquesta manera, es dividirien les diferents tasques i processos entre rols d'usuari i cadascun dels càrrecs podria realitzar la feina que li correspon. La proposta inicial comprèn els rols d'usuari següents:

- Administrador: Introdueix les dades dels plans docents i dóna d'alta Coordinadors i Directors de departament.
- Coordinador: Elabora els horaris.
- Director de departament: Dóna d'alta Responsables de docència.
- Responsable de docència: Dóna d'alta i assigna Professors als grups.
- Professor: Visualitza el seu horari.

A més a més, l'aplicació hauria de ser accessible via web. D'aquesta manera, tots els usuaris podrien utilitzar-la des de qualsevol lloc i dispositiu, sense preocupar-se d'instal·lacions ni actualitzacions.

1.3 Motivacions

En aquesta secció, es parlarà en primera persona sobre quines motivacions personals hi ha darrere del projecte i en justifiquen l'elecció.

Un dels aspectes que més em motiven de la informàtica en general és el fet de poder ajudar la gent a estalviar temps, el qual té un gran valor. En moltes ocasions, una persona, un grup o fins i tot una institució inverteix una quantitat elevada de temps a realitzar certes accions o activitats, sigui en l'àmbit que sigui. Aquest temps es pot reduir si l'acció o activitat en qüestió té una part suficientment mecànica. Tanmateix, encara que no la tingui, sovint és possible desenvolupar una eina informàtica que n'augmenti l'eficiència o, si més no, que la dinamitzi i la faci més còmoda i pràctica.

D'aquí sorgeix el meu objectiu principal en la informàtica, que és precisament aportar el meu gra de sorra a aquesta causa.

Per aquest motiu, m'ha cridat molt l'atenció la proposta d'aquest projecte. És una molt bona oportunitat per contribuir a millorar el procés de gestió i elaboració d'horaris de l'escola, que sol resultar bastant costós en temps per a les persones que hi participen.

D'altra banda, em motiva molt el fet de desenvolupar una aplicació que podrà ser implantada en un entorn real i que podrà beneficiar les persones que la facin servir. No m'atrauria tant dur a terme un altre projecte que, un cop finalitzat, fos simplement arxivat, sense utilitat pràctica per a ningú més que jo, que seria l'únic que me'n beneficiaria, ja que igualment obtindria coneixements i experiència.

Actualment i cada vegada més, m'interessa el desenvolupament d'entorns web. És per això que un factor decisiu a l'hora d'escollir aquesta proposta de PFG ha estat que un dels requisits sigui desenvolupar-lo en format de plataforma web.

Per acabar, m'agradaria afegir que, per motius d'ambició i d'obtenció més àmplia de coneixements, he decidit dur a terme el projecte de manera individual, encara que estigui pensat per a dues persones. Fer-lo sol em suposarà una inversió més gran de temps, però em permetrà no perdre'm cap detall de tot el que suposa desenvolupar una aplicació d'aquest estil.

1.4 Objectius generals

En aquesta secció, s'enumeraran els objectius generals del projecte, que ja s'han deixat entreveure prèviament a la secció 1.2. No obstant això, a continuació se'n presenta la llista completa:

- Desenvolupar una plataforma o aplicació web que permeti dur a terme les tasques de gestió i elaboració dels horaris de l'EPS.

- Proporcionar unes interfícies d'usuari curiosament dissenyades que siguin atractives, intuïtives, còmodes i, sobretot, eficients.
- Emmagatzemar i processar dinàmicament les dades i relacions referents als diversos estudis, cursos, quadrimestres, assignatures, grups, espais, professors, etc.
- Detectar automàticament i avisar de qualsevol tipus d'inconsistència o incompatibilitat horària, per tal d'aportar seguretat al treball.
- Possibilitar la pujada d'arxius externs de dades que serveixin per obtenir la informació bàsica necessària per al funcionament de l'aplicació.
- Permetre la visualització de l'ocupació d'aules, dels horaris dels professors i dels horaris de cada estudi, entre altres vistes que puguin ser d'utilitat per als usuaris.
- Admetre diferents rols d'usuari, als quals s'assigni una sèrie de tasques i un conjunt de permisos concret.

Cal remarcar que la finalitat de l'aplicació no és l'elaboració d'horaris i distribució de docència de manera automàtica. Per contra, el que es vol aconseguir és una eina específica que sigui fàcil i còmoda d'utilitzar i que permeti, gràcies a múltiples ajudes i automatitzacions, realitzar les tasques desitjades manualment.

Un altre punt molt important a tenir en compte és que, a causa de la magnitud dels objectius plantejats, s'ha decidit acotar-los lleugerament: tant la visualització de l'ocupació de les aules com la distribució de docència queden en un segon pla, encara que s'han de tenir molt en compte per tal de facilitar-ne una futura implementació. Això es justifica al capítol 6, en què es conclou que la duració del projecte seria massa gran en cas de no acotar-lo. A més a més, tampoc es preveu el llançament a producció de l'aplicació durant el projecte.

Al capítol 5 es desenvoluparan aquests objectius generals i es concretaran els requisits específics de l'aplicatiu. A més a més, se'n detallarà la prioritat que representen per al projecte.

1.5 Estructura del document

En aquesta secció, es concretaran tant l'estructura del document com les desviacions que presenta respecte del que està estipulat.

El document conté tots els capítols que s'estableixen a la *Guia dels projectes final de grau de l'enginyeria informàtica* [[Comissió dels projectes final de grau de la](#)

UdG 2022].

No obstant això, presenta certes variacions pel que fa a l'ordre. De manera estàndard, els capítols 4, 5 i 6 han de ser, respectivament, “Planificació”, “Marc de treball i conceptes previs” i “Requisits del sistema”.

S’ha considerat que la comprensió del document milloraria si el marc de treball i els requisits es presentessin abans de la planificació, ja que, tant en els requisits com en la planificació, es fan servir la nomenclatura i els conceptes definits en el marc de treball. A més a més, la planificació s’ha realitzat a partir dels requisits.

Així doncs, els capítols 4, 5 i 6 d’aquest document són, respectivament, “Marc de treball i conceptes previs”, “Requisits del sistema” i “Planificació”.

Estudi de viabilitat

En aquest capítol, es presentaran els estudis de viabilitat que feien possible el desenvolupament del projecte. Més concretament, es parlarà de la viabilitat tecnològica (veure secció 2.1) i dels costos de personal que suposaria (veure secció 2.2).

2.1 Viabilitat tecnològica

En aquesta secció, es valorarà si la realització del projecte és tècnicament viable.

Tal com s'ha explicat al capítol 1, el projecte no inclou el llançament a producció del programari. Per tant, com que tracta del desenvolupament d'una aplicació web, només fa falta un ordinador que disposi del *software* necessari per executar-la, de manera local, en l'entorn de desenvolupament (veure capítol 10).

2.2 Costos de personal

En aquesta secció, es calcularan els costos relatius al personal que participaria en el desenvolupament del projecte.

Com que el projecte tracta d'una aplicació web, és necessari diferenciar entre les persones que desenvolupen la part del servidor (*back-end*) i les que desenvolupen la del client (*front-end*).

Tal com es veurà al capítol 7, la tecnologia escollida per al *back-end* és Node.js, mentre que l'escollida per al *front-end* és Vue. Per tant, els preus que es tindran en compte als càlculs són els de desenvolupadors Node.js i desenvolupadors *front-end* JavaScript en general.

A més a més, degut a la importància que prenen per al projecte el disseny de les interfícies d'usuari, cal comptar també amb dissenyadors d'interfícies.

A la taula 2.1 es pot observar el detall de quina és l'aproximació dels costos de personal que suposaria el projecte. El valor dels sous s'ha obtingut de l'informe

de salaris de la *Guia HAYS 2022* [[Hays 2022](#)].

Personal	Temps	Preu	Cost total
Desenvolupadors Node.js	315 hores	18.00 €/hora	5670.00 €
Desenvolupadors <i>front-end</i> JavaScript	440 hores	15.15 €/hora	6666.00 €
Dissenyadors d'interfícies	85 hores	14.68 €/hora	1247.80 €
Total	840 hores		13583.80 €

Taula 2.1: Taula de costos de personal.

El temps de dedicació estimat dels diferents tipus de personal s'ha extret del capítol 6. És important destacar que, per calcular els imports, s'ha fet servir el valor de temps total que comportaria el desenvolupament del 100% del projecte, sense les acotacions marcades al capítol 1.

Finalment, cal esmentar que el temps dedicat a la gestió del projecte s'ha tingut en compte per al càlcul del temps dels desenvolupadors.

Metodologia

En aquest capítol, es presentarà com s'ha efectuat la gestió del projecte (veure secció 3.1) i quina és la metodologia seguida per a desenvolupar-lo (veure secció 3.2).

3.1 Gestió del projecte

En aquesta secció, s'abordaran els criteris seguits en quant a la gestió del projecte que s'han tingut en compte durant tot el seu cicle de vida.

La metodologia de gestió de projectes que s'ha seguit ha estat la que descriu PMBOK. PMBOK (*Project Management Body of Knowledge*) [Project Management Institute 2022] és un llibre que recull un conjunt de terminologia estàndard, processos, bones pràctiques i directrius per a la gestió de projectes. No està restringit ni enfocat a la gestió de projectes d'una àrea en concret, sinó que és bastant genèric.

PMBOK és considerat un gran estàndard pel que fa a la gestió de projectes. Actualment, la majoria de projectes que es duen a terme el segueixen, ja que el fet que sigui tan genèric dóna peu al sorgiment de metodologies de desenvolupament més específiques que l'apliquen i s'hi basen.

D'acord amb PMBOK, el procediment general a seguir a l'hora de desenvolupar un projecte consta de les cinc fases següents:

1. **Iniciació:** moment en què es planteja un problema que requereix una solució. Es defineix en què consistirà el projecte i se n'estudiarà la viabilitat per tal de decidir si tirar-lo endavant.
2. **Planificació:** etapa en què es posen en manifest els objectius que persegueix el projecte i es planifica.
3. **Execució:** recerca i aplicació de solucions que permetin implementar el projecte i satisfer-ne els objectius.
4. **Seguiment i control:** fase que, normalment, s'executa simultàniament

amb l'anterior. Permet controlar i regular el progrés del projecte, anticipar-se a problemes i aprofitar noves oportunitats.

5. Tancament : finalització del projecte. Es comprova que realment se n'hagin assolit els objectius i es presenta als interessats.

3.2 Metodologia de desenvolupament

En aquesta secció, es descriurà quina ha estat la metodologia de desenvolupament que s'ha adoptat a l'hora de planificar i desenvolupar el projecte.

D'entrada, degut a que el projecte és individual, s'han descartat les metodologies AGILE, com ara SCRUM, encara que també siguin útils per a projectes individuals, no s'ha considerat necessària la seva aplicació. Per contra, s'ha optat per una estructuració en paquets de treball anomenada WBS que descriu el propi PMBOK.

WBS (*work-breakdown structure*) és una tècnica o metodologia de desenvolupament basada en la descomposició jeràrquica d'un projecte en el que s'anomenen "paquets de treball". Cada paquet de treball ha de contenir, com a mínim, un identificador, nom, descripció, conjunt de tasques, temporalització i lliurable (més informació al capítol 6). Si no és possible o resulta molt difícil la definició de la temporalització i tasques d'un paquet de treball, significa que es pot i s'ha de descompondre en subpaquets.

Pel que fa al seguiment del projecte, els aspectes que s'han anat desenvolupant han estat revisats pels tutors i valorats en conjunt. S'han realitzat reunions telemàtiques quan s'ha cregut necessari o quan s'han hagut de prendre algunes decisions importants, revisar i valorar algun aspecte o resoldre dubtes.

Marc de treball i conceptes previs

En aquest capítol, es presentaran el marc de treball i els conceptes previs necessaris per fer un seguiment adequat del document i del projecte en general.

Les seccions 4.1 (nomenclatura específica) i 4.2 (funcionament actual) permeten situar el lector en la temàtica que engloba el projecte i entendre la nomenclatura que s'utilitza en el document.

La secció 4.3 (document d'un pla docent) dona a conèixer un dels punts claus del projecte: quines són les dades d'un pla docent que necessita l'aplicació.

Finalment, la secció 4.4 (conceptes tècnics) parla sobre els conceptes tècnics i característiques del tipus d'aplicatiu desenvolupat i permet comprendre la raó de ser de certs aspectes del projecte.

4.1 Nomenclatura específica

En aquesta secció, es definirà la nomenclatura específica del marc de treball del projecte: el procés de gestió dels horaris de la UdG aplicat a l'Escola Politècnica Superior. Aquesta nomenclatura es farà servir en la resta de la memòria.

- **Facultat:** Secció d'una universitat a la qual correspon una branca general de coneixement, com ara "Escola Politècnica Superior" o "Facultat de Medicina" de la UdG.
- **Curs acadèmic:** Període de temps format per l'últim quadrimestre d'un any i el primer del següent.
- **Pla docent:** Instrument que defineix la informació clau sobre els estudis que es cursen en una facultat durant un curs acadèmic. Les dades més rellevants que contenen els de l'EPS són les següents:
 - Per a cada assignatura:
 - * Estudis als quals pertany
 - * Codi

- * Nom
- * Àrees a les quals pertany
- * Departament al qual pertany cadascuna de les àrees
- * Tipus de laboratori que té assignats
- * Crèdits
- * Nombre de grups grans
- * Nombre de grups mitjans
- * Nombre de grups petits
- Per a cada tipus de laboratori:
 - * Nom
 - * Quantitat d'aules
 - * Capacitat d'alumnes
- Departament: Grup destinat a la recerca i a l'ensenyament d'una matèria determinada en una o diverses facultats sota la direcció d'un director, com ara "Informàtica, Matemàtica Aplicada i Estadística" de l'EPS.
- Àrea: Marc d'estudi concret dins de la matèria d'un departament sota la gestió d'un responsable, com ara "Llenguatges i Sistemes Informàtics" del departament d'IMAE.
- Estudi: Terme que engloba els diferents tipus d'estudis universitaris (ja siguin graus, màsters, etc.) com ara el "Grau en Enginyeria Informàtica".
- Curs: Any d'escolaritat d'un estudi que agrupa assignatures repartides en dos quadrimestres (anomenats respectivament primer i segon), com ara "tercer curs" o "quart curs" de GEINF.
- Quadrimestre: Espai de quatre mesos corresponent al final d'un any (primer quadrimestre) o bé a l'inici del següent (segon quadrimestre).
- Setmana: Concepte que fa referència als tipus amb els quals la universitat cataloga les setmanes (A o B). Un quadrimestre està format per setmanes A i per setmanes B. Per aquest motiu, els horaris dels dos tipus de setmana no han de ser necessàriament iguals i un bloc horari pot estar assignat a una setmana concreta o a les dues alhora.
- Vista setmanal: Vista de l'horari d'una setmana concreta o de la combinació de les dues. Les tres vistes possibles són les següents:

- General (conformada per blocs horaris assignats o a una de les setmanes o bé a totes).
- Setmanes A (conformada per blocs horaris assignats a setmanes A).
- Setmanes B (conformada per blocs horaris assignats a setmanes B).
- **Assignatura:** Nom que rep el conjunt de coneixements d'una matèria que es cursa cada any durant un quadrimestre concret, com ara "Estadística". Cada assignatura pertany a una o múltiples àrees i s'imparteix a un o diversos estudis, però no necessàriament en el mateix curs. A més a més, té assignat un conjunt de tipus de laboratori concret.
- **Tipus de laboratori:** Grup de laboratoris que comparteixen les mateixes característiques i finalitats, com ara "Infor". Cada tipus de laboratori té definides tant la quantitat d'aules existent com la capacitat d'alumnes.
- **Aula:** Espai en el qual s'imparteix docència, com ara "PI-01".
- **Grup:** Concepte que es refereix a un grup de qualsevol tipus (gran, mitjà o petit) d'una assignatura determinada, com ara el "grup gran 1" d'Estadística. Una assignatura està composta per un nombre concret de grups de cada tipus i els grups d'un mateix tipus es numeren successivament per tal de distingir-los. A més, en el cas d'una assignatura compartida entre múltiples estudis, cadascun dels seus grups està assignat a un dels estudis o, fins i tot, a més d'un.
- **Bloc horari:** Instància d'un grup determinat assignable a un horari, com ara el "bloc horari 2" del grup gran 1 d'Estadística. S'entén per assignació a un horari el fet de situar-hi un bloc horari en una setmana, dia i hora d'inici concrets. Cadascun conté la informació següent:
 - Grup al qual pertany
 - Dia (en cas que estigui assignat)
 - Hora d'inici (en cas que estigui assignat)
 - Duració
 - Setmana o setmanes
 - Professor que hi imparteix docència
 - Aula en la qual s'imparteix la docència

En la majoria de casos, un grup disposa d'un sol bloc horari, encara que és possible que un grup faci més d'una classe a la setmana i, per tant, hagi

de disposar de més d'un bloc horari.

- **Bloc horari genèric:** Bloc horari que no pertany a un grup, sinó directament a un quadrimestre d'un dels cursos d'un estudi, com ara la "Franja Reservada per a Proves d'Avaluació Continuada" del primer quadrimestre de tercer de GEINF. Cadascun conté la informació següent:
 - Estudi al qual pertany
 - Curs
 - Quadrimestre
 - Etiqueta. Per exemple: "Franja Reservada per a Proves d'Avaluació Continuada"
 - Subetiqueta. Per exemple: "PAC"
 - Dia (en cas que estigui assignat)
 - Hora d'inici (en cas que estigui assignat)
 - Duració
 - Setmana o setmanes
- **Solapament de blocs horaris:** Coincideixen múltiples blocs horaris pertanyents al mateix grup.
- **Solapament de tipus de laboratori:** Excés d'aules d'un tipus de laboratori concret ocupades alhora.
- **Solapament de professors:** Coincideixen múltiples blocs horaris amb el mateix professor assignat.
- **Solapament d'aules:** Coincideixen múltiples blocs horaris amb la mateixa aula assignada.
- **Administrador:** Persona encarregada de gestionar un pla docent i de controlar Coordinadors i Directors de departament.
- **Coordinador:** Persona encarregada de gestionar un estudi i d'elaborar-ne els horaris.
- **Director de departament:** Persona encarregada de dirigir un departament i de controlar els Responsables de docència de les seves àrees.
- **Responsable de docència:** Persona encarregada de gestionar una àrea i de controlar-ne els Professors.
- **Professor:** Persona encarregada d'impartir docència.

4.2 Funcionament actual

En aquesta secció, es detallaran quins són i com funcionen els mètodes de gestió i elaboració d'horaris de l'EPS que s'utilitzen actualment. Es veuran quins càrrecs de l'escola hi participen i les seves funcions, així com la manera en què interactuen entre ells i intercanvien la informació corresponent.

El rectorat reparteix els crèdits a les facultats seguint uns criteris que tenen en compte, entre d'altres aspectes, el nombre d'alumnes matriculats a cada centre i el grau d'experimentalitat dels estudis que imparteix. Cada facultat o escola decideix quin serà el seu pla docent mitjançant la programació d'aquests estudis en funció dels crèdits dels quals disposen.

A partir d'aquest punt, els coordinadors de cada estudi n'elaboren els horaris i l'administrador hi assigna les aules corresponents i comprova la concordança entre el nombre de grups previstos i el nombre de grups que hi apareixen. A més a més, comprova les incompatibilitats horàries que s'hagin pogut produir a causa de l'excés d'ocupació de determinats tipus de laboratori. Un cop s'obtenen uns horaris viables, l'administrador els introdueix a la base de dades del pla docent de la UdG. En aquest moment, els horaris es fan visibles i consultables pel professorat i, en particular, pels responsables de docència. Aquestes persones s'encarreguen d'assignar docència als professors i d'intentar evitar que es produeixin incompatibilitats horàries. És molt comú que, durant aquest procés, es detectin col·lisions entre assignatures que ha d'impartir un mateix professor, fet que implica haver de refer els horaris.

Fins el maig de 2022, aquest procés s'ha fet de manera manual, ja que no hi ha hagut cap eina de suport pel que fa a aquestes tasques. Cada persona dissenyava estratègies pròpies que l'ajudaven a realitzar les seves funcions: des de simples fulls de paper, fins a documents d'Excel o plantilles de PowerPoint. La més sofisticada eren les plantilles de PowerPoint (veure figura 4.1), ja que s'hi dibuixaven les graelles dels horaris, quadres de text per representar blocs horaris, etc. El programa permetia pintar els blocs amb colors per tal de distingir-ne el tipus de grup, saber si era compartit, si tenia restriccions, etc. Encara que era una eina fàcil de fer servir i proporcionava ajuda visual, tenia una gran limitació: no hi havia cap manera fàcil i ràpida de comprovar l'ocupació dels laboratoris, els horaris dels professors, la concordança entre grups compartits, etc.

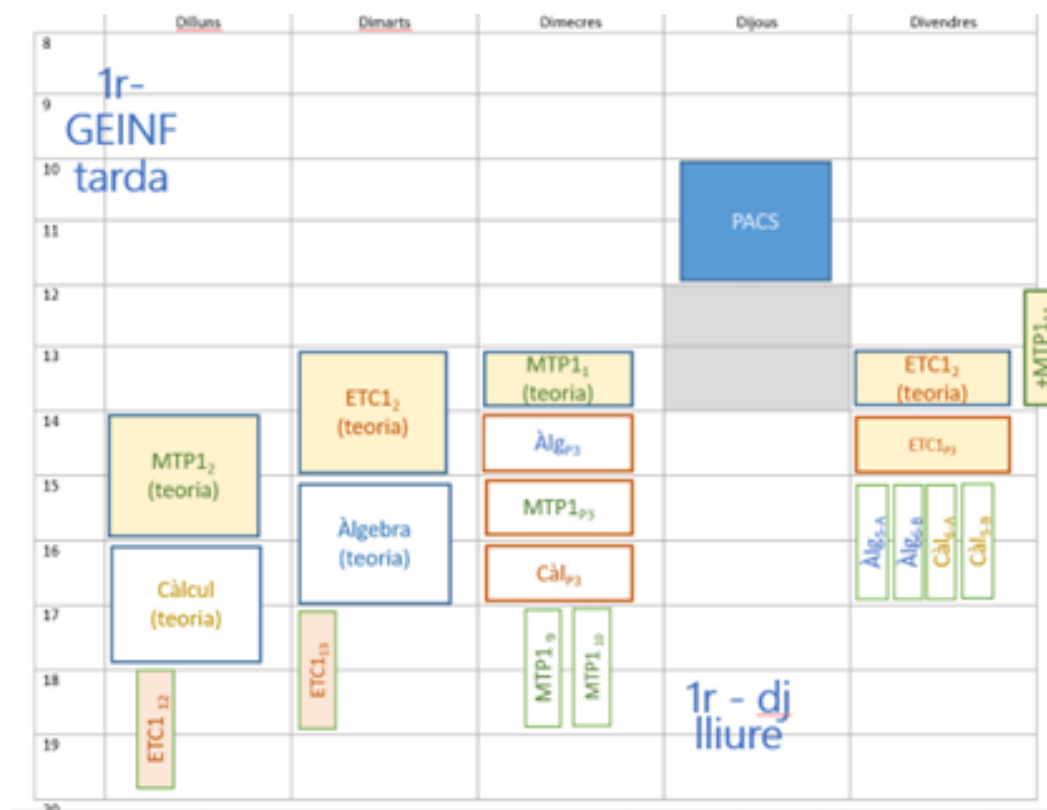


Figura 4.1: Exemple d'una plantilla de PowerPoint utilitzada per a l'elaboració d'horaris.

No obstant això, el maig de 2022, l'equip directiu de l'EPS va posar a disposició dels coordinadors una aplicació que solucionava el problema parcialment. A partir d'una rèplica dels horaris del curs acadèmic actual, afegien o treien grups per tal que s'ajustessin al proper curs. Els grups tenien un estudi, dia, hora d'inici, durada i aula assignada. A més, es podien arrossegar per tal de col·locar-los al lloc corresponent. L'aplicació també comprovava la disponibilitat de les aules i marcava en vermell aquells grups que la seva aula estigués ocupada. Aquesta eina va suposar de gran ajuda per als coordinadors, però, igualment, presentava certes limitacions, com ara:

- Tant les assignatures com els grups no eren fàcilment identificables: calia prémer sobre un grup per poder veure de quin es tractava i a quina assignatura pertanyia.
- Només comprovava la disponibilitat de l'aula en què s'hi havia assignat un grup concret, en comptes de buscar si n'hi havia alguna equivalent disponible.

- No permetia personalitzar la informació que mostrava.
- No es podien afegir fàcilment blocs horaris genèrics per a altres tasques específiques, com ara reunions, proves d'avaluació continuada (PAC), etc.
- No permetia fer el procés d'assignació de docència al professorat ni consultar els horaris dels professors.
- No era fàcilment ampliable ni tampoc una solució integral, ja que només es centrava en una part del procés.

4.3 Document d'un pla docent

En aquesta secció, es descriurà com és el format dels documents utilitzats que contenen la informació d'un pla docent i que, tal com s'ha introduït al capítol 1, l'aplicació ha de poder processar correctament.

Es tracta de fitxers en format *xlsx* (Microsoft Excel) que contenen diversos fulls: un per les assignatures de cada estudi, un per cada agrupació d'assignatures compartides i un per als tipus de laboratori. El nom del full d'un estudi és la seva pròpia abreviació i el de tipus de laboratori és "Laboratoris". Pel que fa al d'una agrupació d'assignatures compartides, no existeix una convenció determinada. Actualment n'hi ha quatre:

- **Comp_ARQ**: Recull les assignatures compartides entre els estudis GARQ i GATE.
- **Compartides**: Recull les assignatures compartides entre els estudis GETI, GEB, GEM, GEE, GEEIA, GEQ.
- **ComAlim**: Recull les assignatures compartides entre els estudis GEA i GINSA.
- **ComInf**: Recull les assignatures compartides entre els estudis GEINF i GDDV.

Cadascun dels fulls que contenen assignatures s'organitzen en forma de taula de manera que les files corresponen a assignatures i les columnes als seus atributs. Els atributs rellevants per a l'aplicació són els següents:

- **Codi**: Codi identificador.
- **Assignatura**: Nom.
- **Àrea**: Àrea a la qual pertany.
- **Dept**: Departament al qual pertany l'àrea.

- Curs: Curs durant el qual s'imparteix.
- Sem: Semestre durant el qual s'imparteix.
- Cr.: Nombre de crèdits.
- NGG: Nombre de grups grans.
- NGM: Nombre de grups mitjans.
- NGP: Nombre de grups petits.
- Labs: Nom del tipus de laboratori al qual s'imparteix.

Existeixen casos en què una assignatura pertany a més d'una àrea i/o s'imparteix a més d'un tipus de laboratori. En aquestes situacions, l'assignatura ocupa més d'una fila amb l'objectiu d'especificar les diferents àrees i tipus de laboratori.

A més a més, és important destacar que una assignatura compartida pot cursar-se a cursos diferents en funció de l'estudi. En aquests casos, els cursos s'indiquen separats per "/". L'ordre d'estudis amb el qual s'especifiquen depèn de quina agrupació d'assignatures compartides es tracti:

- Comp_ARQ: <curs de GARQ>/<curs de GATE>
- Compartides: Totes les assignatures es cursen durant el mateix curs.
- ComAlim: <curs de GEA>/<curs de GINSA>
- ComInf: <curs de GEINF>/<curs de GDDV>

A la figura 4.2 es pot veure l'exemple d'un fragment del full que conté les assignatures de GEINF, mentre que a la figura 4.3 se'n pot veure un del full que conté els tipus de laboratori.

Codi	Assignatura	Dept	Àrea	%	Curs	Sem	Tipus	Cr.	CGG	CGM	CGP	CGPSort	NGG	NGM	NGP	NGPSort	Total	Rend1920	Rend1819	Rend1718	InLabs	Labs
3105G07001	Àlgebra	IMAE	LSI	100	1	1 ob	6	3.5	1.5	1			2	3	6		17.5	0.65	0.65	0.59	24	Infor
3105G07002	Càlcul	IMAE	MA	100	1	1 ob	6	3.5	1.5	1			2	3	6		17.5	0.61	0.61	0.53	24	Infor
3105G07003	Lògica i matemàtica discreta	IMAE	MA	100	1	2 ob	9	4.8	3	1.2			2	3	6		25.8	0.63	0.57	0.52	24	Infor
3105G07006	Física i electrònica	EEEA	ESA	44.44	1	2 ob	9	5.1	1.5	2.4			2	3	9		16.13	0.75	0.53	0.54	15	F3/EB
3105G07006	Física i electrònica	FISICA	FA	55.56	1	2 ob	9	5.1	1.5	2.4			2	3	9		20.2	0.75	0.53	0.54		

Figura 4.2: Fragment del full d'un estudi d'un fitxer de pla docent.

nom	nombre	capacitat
A. Ind	1	16
AC239	1	15
Agr	1	16
ASTRID	1	0
Agr-PV/Hiver Tallers	1	16
Anal.Q.	1	15
aula	100	24
aula arq.	1	20
Bioq.i E.	1	16
C.i F.Ind	1	16

Figura 4.3: Fragment del full dels tipus de laboratori d'un fitxer de pla docent.

4.4 Conceptes tècnics

4.4.1 Introducció

En aquesta subsecció, s'introduiran els temes i els conceptes que es tractaran a la resta de la secció.

Tal com s'ha esmentat al capítol 1, la solució informàtica al problema que planteja el projecte s'ha d'implementar en format de plataforma o aplicació web. Si aquest requisit es suma al fet que l'emmagatzematge de dades és totalment necessari, ràpidament es pot concloure que l'arquitectura del sistema és de naturalesa client-servidor i que, consegüentment, es divideix en dues parts diferenciades (veure figura 4.4): l'aplicació client o *front-end* (veure subsecció 4.4.2) i l'aplicació servidor o *back-end* (veure subsecció 4.4.3).

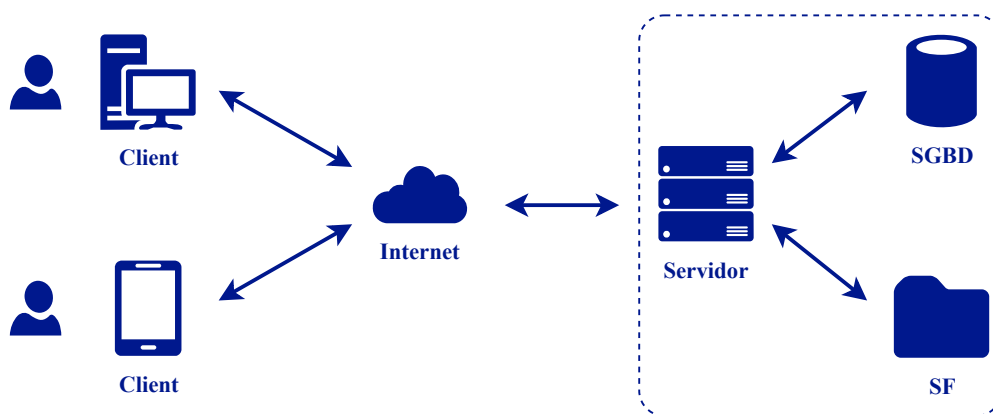


Figura 4.4: Esquema de l'arquitectura d'una aplicació web convencional.

L'aplicació client s'executa sobre un navegador web i s'encarrega principalment d'interactuar amb l'usuari, de manera que pugui visualitzar i manipular les dades emmagatzemades al servidor. Per poder accedir a aquestes dades, l'aplicació

client ha de ser capaç de comunicar-se a través d'Internet amb l'aplicació servidor, ja que s'executa a la màquina de l'usuari (veure subsecció 4.4.4).

L'aplicació servidor s'executa fora del navegador en una màquina remota. S'encarrega principalment d'atendre i servir les peticions de consulta o manipulació de dades que rep de les aplicacions client. Davant de cada petició, l'aplicació executa la lògica de negoci corresponent i persisteix les dades necessàries al SF (sistema de fitxers) o a la base de dades, la qual és operada per un SGBD (sistema gestor de bases de dades) (veure subsecció 4.4.5). A més a més, a part de simplificar el desenvolupament de l'aplicació client, impedeix l'accés directe a la base de dades des d'Internet, fet que seria molt perillós.

4.4.2 Aplicació client

En aquesta subsecció, es donaran a conèixer les tecnologies involucrades en el desenvolupament d'aplicacions web client.

Les aplicacions web s'executen sobre un navegador i, generalment, estan compostes pels tres elements bàsics següents:

- HTML (*HyperText Markup Language*) [WHATWG 2022]: Llenguatge de marcat que defineix la semàntica i l'estructura del contingut d'una pàgina web.
- CSS (*Cascade Style Sheets*) [W3C 2022a]: Llenguatge de fulls d'estil que defineix els estils i la presentació d'un document HTML.
- JS (*JavaScript*) [MDN Web Docs 2022]: Llenguatge de programació que defineix la lògica i el comportament del contingut d'una pàgina web i la fa interactiva. Es tracta d'un llenguatge d'un sol fil d'execució, que té suport per a programació orientada a objectes, imperativa i declarativa i que pot ser interpretat o compilat en temps d'execució.

Per poder desenvolupar una aplicació web interactiva i dinàmica, en la gran majoria dels casos és necessari treballar amb el DOM mitjançant JavaScript.

El DOM (*Document Object Model*) és una interfície per a documents HTML que n'estructura els elements en forma d'arbre i permet crear-hi, modificar-hi o esborrar-hi nodes, els quals representen un element HTML determinat. A través d'aquesta interfície és possible mutar en temps real el document HTML que el navegador està presentant.

Treballar directament i de manera adequada amb el DOM requereix una comprensió molt profunda del seu funcionament. A mesura que augmenta el dinamisme de l'aplicació i s'implementen interaccions més complexes, el seu desen-

volupament es complica notablement i la seva mantenibilitat disminueix. A més a més, requereix una quantitat de temps considerable.

Degut a aquests motius, eventualment van sorgir els *frameworks* JavaScript amb l'objectiu de brindar una millor experiència de desenvolupament i augmentar la mantenibilitat i l'escalabilitat de les aplicacions web.

Un *framework* JavaScript és una infraestructura de *software* que proporciona un conjunt d'eines implementades i provades que serveixen de base per crear aplicacions interactives i escalables. Els *frameworks* abstreuen la gestió del DOM per tal d'estalviar al desenvolupador el fet d'haver de manipular-lo directament en la majoria d'ocasions.

Actualment, existeixen un munt de *frameworks* per escollir. La majoria funcionen de manera similar, però realitzen algunes tasques de manera diferent. Cadascun té els seus avantatges i inconvenients i pot estar enfocat a un tipus d'aplicació concret, cosa que provoca que no n'existeixi un de definitiu. A la pràctica, l'opció més viable sempre depèn del tipus d'aplicació que s'hagi d'implementar. Avui dia, els més populars són React JS, Vue JS, Angular JS i Ember JS, entre altres.

Quan es treballa amb aquest tipus de *software* i altres biblioteques de codi, de seguida apareix la necessitat d'unes eines anomenades *bundlers* o “empaquetadors”.

Abans que els mòduls ES (que permeten la importació i exportació de fragments de codi JavaScript) estéssin disponibles per a navegadors web, els desenvolupadors no tenien cap mecanisme natiu per crear codi JavaScript de manera modular. Per aquest motiu, l'ús de *bundlers* sempre havia sigut necessari per tal de concatenar el codi de *frameworks* i biblioteques en arxius que poguéssin ser executats en el navegador. WebPack, Rollup i Parcel són uns dels *bundlers* més populars.

4.4.3 Aplicació servidor

En aquesta subsecció, es donaran a conèixer les tecnologies involucrades en el desenvolupament d'aplicacions servidor.

A diferència de les aplicacions web client, les aplicacions servidor no estan tan limitades pel que fa a llenguatges de programació. Pràcticament es poden desenvolupar fent servir qualsevol llenguatge. No obstant això, tampoc es lliuren de la necessitat de l'existència de *frameworks* que proporcionin eines que rellevin el desenvolupador en les tasques més comunes, complicades o de baix nivell.

Actualment, hi ha una gran varietat d'opcions a l'hora d'escollir la tecnologia amb la qual s'ha de desenvolupar una determinada aplicació servidor. La tria depèn de múltiples factors, com ara quina és la càrrega de peticions esperada o quina importància té el rendiment envers altres aspectes. Avui dia, els més populars són Firebase, Express.js, Spring Boot, Laravel, Django i Ruby on Rails, entre altres.

4.4.4 Comunicació entre client i servidor

En aquesta subsecció, es parlarà sobre com s'estableix la comunicació entre l'aplicació client i l'aplicació servidor.

Tal com s'ha pogut veure anteriorment en la secció, l'existència d'eines que permetin implementar la comunicació entre diferents aplicacions és completament essencial. Aquestes eines reben el nom d'API (*application programming interface*) i consisteixen un conjunt de normes i protocols que permeten la comunicació entre dues aplicacions, la qual es realitza mitjançant peticions i respostes.

En el cas de la comunicació entre el *front-end* i el *back-end* d'una aplicació web, es necessiten API enfocades a l'obtenció i manipulació de dades. Al llarg del temps, han sorgit diversos protocols i arquitectures que permeten desenvolupar aquesta tipologia d'API, com ara XML-RPC, SOAP, JSON-RPC, REST o GraphQL.

Actualment, l'opció més popular i utilitzada pel desenvolupament d'aquest tipus d'API és REST (*Representational State Transfer*) [Red Hat 2022]. A diferència d'altres opcions com SOAP, REST no és ni un protocol ni un estàndard, sinó un conjunt de pautes pel que fa a l'arquitectura. Una API és considerada una "RESTful API" o "API REST" si satisfà tots els principis REST [Lokesh Gupta 2022].

Tal com el seu nom indica, quan el client llança una petició contra una API REST, l'API li transfereix una representació de l'estat actual del recurs demanat. La informació s'entrega mitjançant el protocol HTTP (*Hypertext Transfer Protocol*) [W3C 2022b] i en format JSON, XML o HTML, entre altres. El format més usat és JSON (*JavaScript Object Notation*) [JSON 2022], ja que és fàcilment comprensible tant per màquines com persones i, a més, no depèn de cap llenguatge de programació, encara que el seu nom pugui indicar el contrari.

4.4.5 Emmagatzematge i gestió de dades

En aquesta subsecció, es parlarà sobre com es gestiona l'emmagatzematge de les dades a la banda del servidor.

En els últims anys, les dades s'han convertit en un dels punts més crítics de qualsevol aplicació. La importància d'una bona administració de les dades és clau i beneficiosa tant per a l'usuari del *software* com per a l'organització que l'ha desenvolupat.

No obstant això, aconseguir portar-ho a la pràctica no és tasca senzilla. Aquí és on entren en joc els SGBD (sistemes gestors de bases de dades). Un SGBD és un tipus de programari dedicat a fer d'interfície entre la base de dades i els seus usuaris. Permet administrar, emmagatzemar i recuperar la informació d'una base de dades de manera pràctica, eficient i segura. A més a més, poden implementar funcionalitats addicionals molt útils, com ara visualització de les dades o replicació.

Generalment s'accedeix a les dades mitjançant llenguatges de consulta, com ara SQL (*Structured Query Language*).

Una de les característiques més rellevants dels SGBD és que han de complir sí o sí tots els principis ACID (*Atomicity, Consistency, Isolation, Durability*). Això garanteix la fiabilitat de les seves transaccions, enteses com operacions compostes per altres que són individuals. Més concretament, els principis ACID són els següents:

- **Atomicitat:** D'una transacció o se n'executen totes les operacions que la conformen o bé no se n'executa cap.
- **Consistència:** L'estat en el qual es troba la base de dades quan finalitza una transacció ha de ser coherent amb l'estat que tenia abans de que comencés.
- **Aïllament:** Cada transacció s'ha d'executar de manera aïllada envers la resta d'operacions.
- **Definitivitat:** Si es confirma una transacció, el resultat ha de ser definitiu i no es pot perdre.

De nou, l'elecció d'un SGBD depèn en gran mesura de l'aplicació que s'hagi de desenvolupar. Existeixen un munt d'alternatives enfocades a diferents propòsits: tractament de dades relacionades, gestió de grans volums de dades, consulta molt ràpida, cerques eficients, etc.

Actualment, els SGBD més populars són Oracle, MySQL, PostgreSQL, Redis, Elasticsearch i MongoDB, entre altres.

Requisits del sistema

En aquest capítol, es presentaran quins són els requisits que ha de complir el sistema. Més concretament, a la secció 5.2, es detallaran quins són els requisits funcionals, mentre que a les seccions 5.3 i 5.4 s'indicaran, respectivament, quins són els no funcionals i els de domini. No obstant això, abans de res, es definirà l'estructura dels requisits i es realitzaran una sèrie de consideracions inicials (veure secció 5.1). Finalment, a la secció 5.5 es representarà la matriu de dependències dels requisits.

5.1 Consideracions inicials

En aquesta secció, es presentaran una sèrie de consideracions inicials amb l'objectiu de clarificar el contingut de les seccions que segueixen.

En primer lloc, és necessari definir el format que adoptarà cadascun dels requisits del sistema:

Tipus-Numeració [Prioritat]: Descripció

Més concretament, el tipus de requisit es representarà mitjançant les seves sigles, com ara RF per als funcionals o RNF per als no funcionals. La numeració es dividirà en grups en funció del rol d'usuari al qual pertanyi el requisit i inclourà la inicial del nom del rol. Pel que fa a la prioritat, se n'han definit tres nivells:

- **Prioritat [1] o essencial:** el compliment total del requisit és de vital importància.
- **Prioritat [2] o moderada:** és important haver desenvolupat, almenys, una part del requisit.
- **Prioritat [3] o addicional:** el compliment del requisit no té gaire rellevància, ja que està més enfocat al treball futur.

D'altra banda, cal remarcar que les descripcions dels requisits utilitzen la nomenclatura específica del marc de treball, detallada al capítol 4.

A més a més, per evitar explicacions redundants, d'ara en endavant, quan es

parli de la visualització dels horaris d'un estudi, no s'estarà fent referència a una vista del conglomerat d'horaris de tots els seus cursos i quadrimestres, sinó de vistes en què es mostra l'horari d'un quadrimestre específic en un curs específic. De la mateixa manera, la visualització dels horaris d'un professor o la de l'ocupació d'una aula es separa en quadrimestres.

5.2 Requisits funcionals

En aquesta secció, es presentaran els requisits funcionals que ha de complir el programari del projecte. Tal com s'ha pogut veure tant en el capítol 1 com en altres, el sistema involucra diversos rols d'usuari. Cada rol ha de poder realitzar un conjunt d'accions determinades i, consegüentment, cadascun comporta una sèrie de requisits diferents.

Per aquest motiu, els requisits funcionals s'han estructurat en funció d'aquests rols: la subsecció 5.2.2 per als Administradors, la subsecció 5.2.3 per als Coordinadors, la subsecció 5.2.4 per als Directors de departament, la subsecció 5.2.5 per als Responsables de docència i la subsecció 5.2.6 per als Professors.

A banda d'això, com és natural, també hi ha requisits comuns per a tots els rols, els quals s'agrupen a la subsecció 5.2.1.

5.2.1 Requisits generals

En aquesta subsecció, es llistaran els requisits funcionals generals del sistema, comuns per a tots els usuaris, independentment dels seus rols.

- **RF-G1 [1]**: Assegurar l'autenticació de tots els usuaris a través d'un formulari de *login*, que s'ha de mostrar a la pantalla quan un usuari no autenticat accedeix a l'aplicació. Sense estar-ho, no l'ha de poder fer servir. L'autenticació d'un usuari ha de suposar la generació d'un *JSON Web Token* [IETF 2022] (*token* a partir d'ara), per tal de mantenir la seva sessió i poder ser identificat de forma segura pel procés d'autorització de l'aplicació de l'API (o servidor).
- **RF-G2 [1]**: No permetre l'autoregistre d'usuaris, ja que usuaris de determinats rols s'encarregaran de donar d'alta altres usuaris del rol que els correspongui. El procés d'alta d'usuaris es detalla a continuació:
 1. L'usuari que registra introdueix les dades de l'usuari que vol donar d'alta, entre les quals consta la seva adreça de correu electrònic. A continuació, l'usuari es crea però amb l'estat de "desactivat".

2. El nou usuari rep un *email* de confirmació amb un enllaç. Mentrestant, no pot autenticar-se a l'aplicació, ja que encara està desactivat.
 3. Un cop l'usuari accedeix a l'enllaç, pot procedir a crear la seva contrasenya mitjançant el formulari presentat a la pantalla. La nova contrasenya s'envia, juntament amb el *token* que conté l'enllaç, al servidor. Si el *token* és vàlid i no ha expirat, l'operació es farà efectiva. La contrasenya s'emmagatzema encriptada.
- **RF-G3 [1]**: Posar a disposició de l'usuari l'opció de restablir, en cas de pèrdua, la seva contrasenya des de la pàgina de *login*. El procediment ha d'utilitzar el seu correu electrònic com a punt de recuperació i ha de funcionar de manera segura a través d'un *token*.
 - **RF-G4 [1]**: L'aplicació del servidor ha d'integrar un mecanisme d'autorització per tal de restringir l'accés als *endpoints* de l'API. Només han de poder utilitzar-los els usuaris que s'hagin autenticat prèviament i que, a més, tinguin permís per fer-ho, de manera que cada *endpoint* sigui accessible només per a un conjunt d'usuaris concret.
 - **RF-G5 [1]**: Mostrar en tot moment el nom i el rol de l'usuari autenticat, per tal d'evitar confusions.
 - **RF-G6 [1]**: Mostrar un menú que permeti a l'usuari navegar entre les diferents pàgines que exposen les funcionalitats principals corresponents al seu rol.
 - **RF-G7 [1]**: Els registres de la base de dades no s'han d'esborrar definitivament. En comptes d'això, tots han de tenir un camp que guardi la data en què han estat eliminats, si és que s'han eliminat. Si no ho estan, aquest camp ha de ser nul.

5.2.2 Requisits dels Administradors

En aquesta subsecció, es llistaran els requisits funcionals particulars dels usuaris amb rol d'Administrador. Cadascun dels apartats que segueixen correspon a una de les seves funcionalitats principals: gestió del pla docent, assignació d'aules, gestió de Coordinadors i gestió de Directors de departament.

5.2.2.1 Gestió del pla docent

- **RF-A1 [1]**: Carregar el pla docent del curs acadèmic següent, de manera que es mantinguin totes les dades que no hagin canviat respecte de l'actual.

- **RF-A2 [1]**: Visualitzar les dades del pla docent: estudis i assignatures, departaments i àrees, tipus de laboratori, etc.
- **RF-A3 [1]**: Modificar les dades del pla docent que es llisten a continuació:
 - De les assignatures: nombre de grups de cada tipus i àrees i tipus de laboratori assignats.
 - Dels tipus de laboratori: quantitat d'aules i capacitat d'alumnes.

5.2.2.2 Assignació d'aules

- **RF-A4 [3]**: Assignar una aula concreta a cada bloc horari que pugui realitzar-se a més d'una, a mesura que els Coordinadors acabin d'elaborar els horaris que els pertocuin.

5.2.2.3 Gestió de Coordinadors

- **RF-A5 [1]**: Visualitzar la llista de les assignacions de Coordinador als diferents estudis.
- **RF-A6 [1]**: Assignar un dels Coordinadors encara no assignats a qualsevol dels estudis.
- **RF-A7 [1]**: Quan es carrega un nou pla docent, s'han de mantenir les assignacions de Coordinador als estudis que ja hi havia a l'anterior.
- **RF-A8 [1]**: Desassignar un Coordinador ja assignat a un estudi.
- **RF-A9 [1]**: Visualitzar el llistat de tots els usuaris Coordinadors, juntament amb la seva informació: nom complet, adreça de correu electrònic, grau al qual està assignat aquest curs acadèmic i si està o no activat.
- **RF-A10 [1]**: Donar d'alta usuaris Coordinadors, és a dir, usuaris amb rol de Coordinador. Per fer-ho, ha d'entrar el seu nom, cognoms i adreça de correu electrònic. A continuació, es duu a terme el procés genèric explicat a la subsecció 5.2.1. A més a més, ha de poder reenviar-li el correu d'activació.
- **RF-A11 [1]**: Donar de baixa usuaris Coordinadors.

5.2.2.4 Gestió de Directors de departament

- **RF-A12 [1]**: Visualitzar la llista de les assignacions de Director de departament als diferents departaments.

- **RF-A13 [1]:** Assignar un dels Directors encara no assignats a qualsevol dels departaments.
- **RF-A14 [1]:** Quan es carrega un nou pla docent, s'han de mantenir les assignacions de Director als departaments que ja hi havia a l'anterior.
- **RF-A15 [1]:** Desassignar un Director ja assignat a un departament.
- **RF-A16 [1]:** Visualitzar el llistat de tots els usuaris Directors de departament, juntament amb la seva informació: nom complet, adreça de correu electrònic, departament al qual està assignat aquest curs acadèmic i si està o no activat.
- **RF-A17 [1]:** Donar d'alta usuaris Directors de departament, és a dir, usuaris amb rol de Director de departament. Per fer-ho, ha d'entrar el seu nom, cognoms i adreça de correu electrònic. A continuació, es duu a terme el procés genèric explicat a la subsecció 5.2.1. A més a més, ha de poder reenviar-li el correu d'activació.
- **RF-A18 [1]:** Donar de baixa usuaris Directors de departament.

5.2.3 Requisits dels Coordinadors

En aquesta subsecció, es llistaran els requisits funcionals particulars dels usuaris amb rol de Coordinador. Cadascun dels apartats que segueixen correspon a una de les seves funcionalitats principals: gestió d'horaris d'estudis, consulta d'horaris de professors i consulta d'horaris d'aules.

5.2.3.1 Gestió d'horaris d'estudis

- **RF-C1 [1]:** Visualitzar els horaris de l'estudi que gestiona, separats en cursos i quadrimestres.
- **RF-C2 [1]:** Visualitzar els horaris dels estudis que ofereixin alguna assignatura compartida amb l'estudi que gestiona.
- **RF-C3 [1]:** Modificar els horaris dels quadrimestres de qualsevol dels cursos de l'estudi que gestiona. El ventall de possibilitats que ha de proporcionar el procés d'edició es detalla a continuació:
 - Alternar entre les diferents vistes setmanals, descrites al capítol 4.
 - Consultar els blocs horaris pendents de col·locar de manera agrupada per assignatures i tipus de grup.

- Consultar la informació completa de qualsevol bloc horari o bloc horari genèric.
- Modificar la informació temporal (dia, hora d'inici, duració i setmana) dels blocs horaris i dels blocs horaris genèrics.
- Modificar les assignacions a estudis del grup de qualsevol bloc horari l'assignatura del qual sigui compartida.
- Modificar l'etiqueta i la subetiqueta dels blocs horaris genèrics.
- Oferir la possibilitat d'arrossegar els blocs horaris i els blocs horaris genèrics per tal de modificar-ne la informació temporal.
- Oferir la possibilitat d'allargar i escurçar els blocs horaris i els blocs horaris genèrics per tal de modificar-ne l'hora d'inici i la duració.
- Crear o esborrar blocs horaris i blocs horaris genèrics.
- Aplicar un filtratge de blocs horaris d'assignatures compartides el grup dels quals no estigui assignat a cap estudi.
- Aplicar un filtratge de blocs horaris d'assignatures compartides el grup dels quals estigui assignat només a altres estudis.
- Quan s'arrossegui un bloc horari, visualitzar una marca sobre les franges en què es produïria algun tipus de solapament en cas que s'hi col·loqués.
- Visualitzar una marca sobre els blocs horaris que provoquin solapaments.
- Activar o desactivar la visualització de solapaments de cada tipus.
- Consultar la informació detallada dels solapaments: veure quins cursos de cada estudi hi estan implicats.

5.2.3.2 Consulta d'horaris de Professors

- **RF-C4 [2]**: Visualitzar els horaris dels Professors que imparteixin docència a alguna de les assignatures que ofereixi el grau que gestiona.

5.2.3.3 Consulta d'horaris d'aules

- **RF-C5 [3]**: Visualitzar els horaris de l'ocupació de qualsevol aula de l'escola.

5.2.4 Requisits dels Directors de departament

En aquesta subsecció, es llistaran els requisits funcionals particulars dels usuaris amb rol de Director de departament. Cadascun dels apartats que segueixen correspon a una de les seves funcionalitats principals: consulta d'horaris de Professors, consulta d'horaris d'estudis i gestió de Responsables de docència.

5.2.4.1 Consulta d'horaris de Professors

- **RF-D1 [3]**: Visualitzar els horaris dels Professors que imparteixin docència a alguna assignatura de qualsevol de les àrees del seu departament.

5.2.4.2 Consulta d'horaris d'estudis

- **RF-D2 [2]**: Visualitzar els horaris dels estudis que ofereixin alguna assignatura de qualsevol de les àrees del seu departament.

5.2.4.3 Gestió de Responsables de docència

- **RF-D3 [1]**: Visualitzar la llista de les assignacions de Responsable de docència a les diferents àrees del seu departament.
- **RF-D4 [1]**: Assignar un dels Responsables de docència encara no assignats a qualsevol de les àrees del seu departament.
- **RF-D5 [1]**: Quan es carrega un nou pla docent, s'han de mantenir les assignacions de Responsable de docència a les àrees del seu departament que ja hi havia a l'anterior.
- **RF-D6 [1]**: Desassignar un Responsable de docència ja assignat a una àrea.
- **RF-D7 [1]**: Visualitzar el llistat de tots els usuaris Directors de departament, juntament amb la seva informació: nom complet, adreça de correu electrònic, àrea a la qual està assignat i si està o no activat.
- **RF-D8 [1]**: Donar d'alta usuaris Responsables de docència, és a dir, usuaris amb rol de Responsable de docència. Per fer-ho, ha d'entrar el seu nom, cognoms i adreça de correu electrònic. A continuació, es duu a terme el procés genèric explicat a la subsecció 5.2.1. A més a més, ha de poder reenviar-li el correu d'activació.
- **RF-D9 [1]**: Donar de baixa usuaris Responsables de docència.

5.2.5 Requisits dels Responsables de docència

En aquesta subsecció, es llistaran els requisits particulars dels usuaris amb rol de Responsable de docència. Cadascun dels apartats que segueixen correspon a una de les seves funcionalitats principals: consulta d'horaris de Professors, assignació de Professors i gestió de Professors.

5.2.5.1 Consulta d'horaris de Professors

- **RF-R1 [3]**: Visualitzar els horaris dels Professors que imparteixin docència a alguna assignatura de la seva àrea.

5.2.5.2 Assignació de Professors

- **RF-R2 [2]**: Assignar un Professor a cada bloc horari de cadascun dels grups de les assignatures de la seva àrea.
- **RF-R3 [2]**: Visualitzar una marca sobre els blocs horaris que provoquin solapaments de Professors.

5.2.5.3 Gestió de Professors

- **RF-R4 [3]**: Visualitzar el llistat de tots els usuaris Professors que hagin d'impartir docència a alguna de les assignatures de la seva àrea, juntament amb la seva informació: nom complet, adreça de correu electrònic, assignatures en què imparteix docència i si està o no activat.
- **RF-R5 [3]**: Donar d'alta usuaris Professors, és a dir, usuaris amb rol de Professor. Per fer-ho, ha d'entrar el seu nom, cognoms i adreça de correu electrònic. A continuació, es duu a terme el procés genèric explicat a la subsecció 5.2.1. A més a més, ha de poder reenviar-li el correu d'activació.
- **RF-R6 [3]**: Donar de baixa usuaris Professors.

5.2.6 Requisits dels Professors

En aquesta subsecció, es llistaran els requisits funcionals particulars dels usuaris amb rol de Professor. Cadascun dels apartats que segueixen correspon a una de les seves funcionalitats principals: consulta d'horaris propis, consulta d'horaris d'assignatures, consulta d'horaris d'estudis.

5.2.6.1 Consulta d'horaris propis

- **RF-P1 [3]**: Visualitzar els seus propis horaris, és a dir, els blocs horaris que li han estat assignats.

5.2.6.2 Consulta d'horaris d'assignatures

- **RF-P2 [3]**: Visualitzar els horaris de les assignatures a les quals imparteixi docència.

5.2.6.3 Consulta d'horaris d'estudis

- **RF-P3 [3]**: Visualitzar els horaris dels estudis que ofereixin alguna de les assignatures en què imparteixi docència.

5.3 Requisits no funcionals

En aquesta secció, es llistaran els requisits no funcionals del sistema.

- **RNF-1 [1]**: El client de l'aplicació ha d'executar-se sobre un entorn web. S'ha de poder comunicar amb el procés del servidor mitjançant crides HTTP a l'API del servidor.
- **RNF-2 [1]**: La part de gestió i persistència de dades ha de ser administrada per l'aplicació del servidor, que ha d'exposar una RESTful API. Aquest procés ha de comunicar-se amb els Sistemes Gestors de Bases de Dades escollits per allotjar la informació del projecte.
- **RNF-3 [1]**: Les interfícies d'usuari han d'estar curosament dissenyades i han de ser atractives, intuïtives i, sobretot, eficients.

5.4 Requisits de domini

En aquesta secció, es llistaran els requisits de domini del sistema.

- **RD-1 [1]**: El format del fitxer que conté les dades del pla docent d'un curs acadèmic està fixat per l'escola (veure capítol 4).

5.5 Matriu de dependències

En aquesta secció, es representarà la matriu de dependències dels requisits. D'aquesta manera, es podran identificar els requisits funcionals que necessitin l'im-

plementació d'altres per poder-se dur a terme.

El format que adoptarà cadascuna de les dependències es defineix a continuació:

$$(\mathbf{RF-X}, \mathbf{RF-Y}, \dots) \longrightarrow (\mathbf{RF-P}, \mathbf{RF-Q}, \dots)$$

És a dir, la llista de requisits de la part anterior de la fletxa depèn de la de la part posterior.

La matriu de dependències completa és la següent:

- $(\mathbf{RF-G5}, \mathbf{RF-G6}) \longrightarrow (\mathbf{RF-G1}, \mathbf{RF-G2})$
- $(\mathbf{RF-A2}, \mathbf{RF-A3}, \mathbf{RF-A4}, \mathbf{RF-A7}, \mathbf{RF-A10}, \mathbf{RF-A14}, \mathbf{RF-A17}) \longrightarrow (\mathbf{RF-A1})$
- $(\mathbf{RF-C5}) \longrightarrow (\mathbf{RF-A4})$
- $(\mathbf{RF-A5}, \mathbf{RF-A8}, \mathbf{RF-C3}, \mathbf{RF-C5}) \longrightarrow (\mathbf{RF-A6})$
- $(\mathbf{RF-A7}, \mathbf{RF-A9}, \mathbf{RF-A11}) \longrightarrow (\mathbf{RF-A10})$
- $(\mathbf{RF-A12}, \mathbf{RF-A15}) \longrightarrow (\mathbf{RF-A13})$
- $(\mathbf{RF-A13}, \mathbf{RF-A16}, \mathbf{RF-A18}) \longrightarrow (\mathbf{RF-A17})$
- $(\mathbf{RF-A4}, \mathbf{RF-C1}, \mathbf{RF-C2}, \mathbf{RF-D2}, \mathbf{RF-R1}, \mathbf{RF-R2}) \longrightarrow (\mathbf{RF-C3})$
- $(\mathbf{RF-D3}, \mathbf{RF-D6}) \longrightarrow (\mathbf{RF-D4})$
- $(\mathbf{RF-D4}, \mathbf{RF-D7}, \mathbf{RF-D9}) \longrightarrow (\mathbf{RF-D8})$
- $(\mathbf{RF-C4}, \mathbf{RF-D1}, \mathbf{RF-R3}, \mathbf{RF-P1}, \mathbf{RF-P2}, \mathbf{RF-P3}) \longrightarrow (\mathbf{RF-R2})$
- $(\mathbf{RF-R4}, \mathbf{RF-R6}) \longrightarrow (\mathbf{RF-R5})$

Planificació

En aquest capítol, es presentarà la planificació del projecte. Més concretament, se'n recolliran les tasques que involucra (veure secció 6.1) i tot seguit, s'utilitzaran per elaborar-ne una temporalització (veure secció 6.3).

6.1 Descomposició i planificació de tasques

En aquesta secció, es presentarà la descomposició i la planificació de les tasques que involucra el projecte.

Primerament, es definirà un esquema arbòric que representi les diferents parts que conformen el projecte des d'una visió generalista. Cadascuna d'aquestes parts conformarà una subsecció. Tot seguit, s'aprofundirà en cadascuna de les branques d'aquest arbre general i es determinaran els diversos paquets de treball que les formen. Degut a la grandària d'alguna de les branques, és possible que en alguns casos siguin necessàries més descomposicions.

6.1.1 Consideracions inicials

En aquesta subsecció, es realitzaran una sèrie de consideracions inicials amb l'objectiu de clarificar el contingut de la resta de la secció.

Tal com s'ha vist al capítol 3, la gestió del projecte es basa en el PMBOK. Més concretament, en la metodologia de desenvolupament WBS, a través de la qual es durà a terme una descomposició del projecte en paquets de treball per tal d'organitzar i planificar totes les tasques que s'hi involucren.

Cada paquet de treball contindrà la informació següent:

- **Identificador:** Codi que servirà per referir-se al paquet en seccions posteriors.
- **Nom:** Nom que se li haurà donat al paquet als esquemes arbòrics.
- **Descripció:** Descripció breu del paquet.
- **Tasques:** Conjunt de tasques que s'han de realitzar en el paquet.

- **Temporalització:** Estimació del temps que es necessitarà per completar les tasques del paquet.
- **Lliurable:** Resultat palpable que s'obtindrà un cop completades les tasques del paquet.

És molt important destacar que la planificació s'ha fet tenint en compte el 100% del projecte, sense les acotacions marcades al capítol 1. Gràcies a això, es pot observar que sense aquestes acotacions la data de finalització estimada del projecte se'n va massa enllà (veure secció 6.3).

6.1.2 Visió general

En aquesta subsecció, es mostrarà la descomposició general de les tasques del projecte en branques, tal com es pot veure a la figura 6.1.

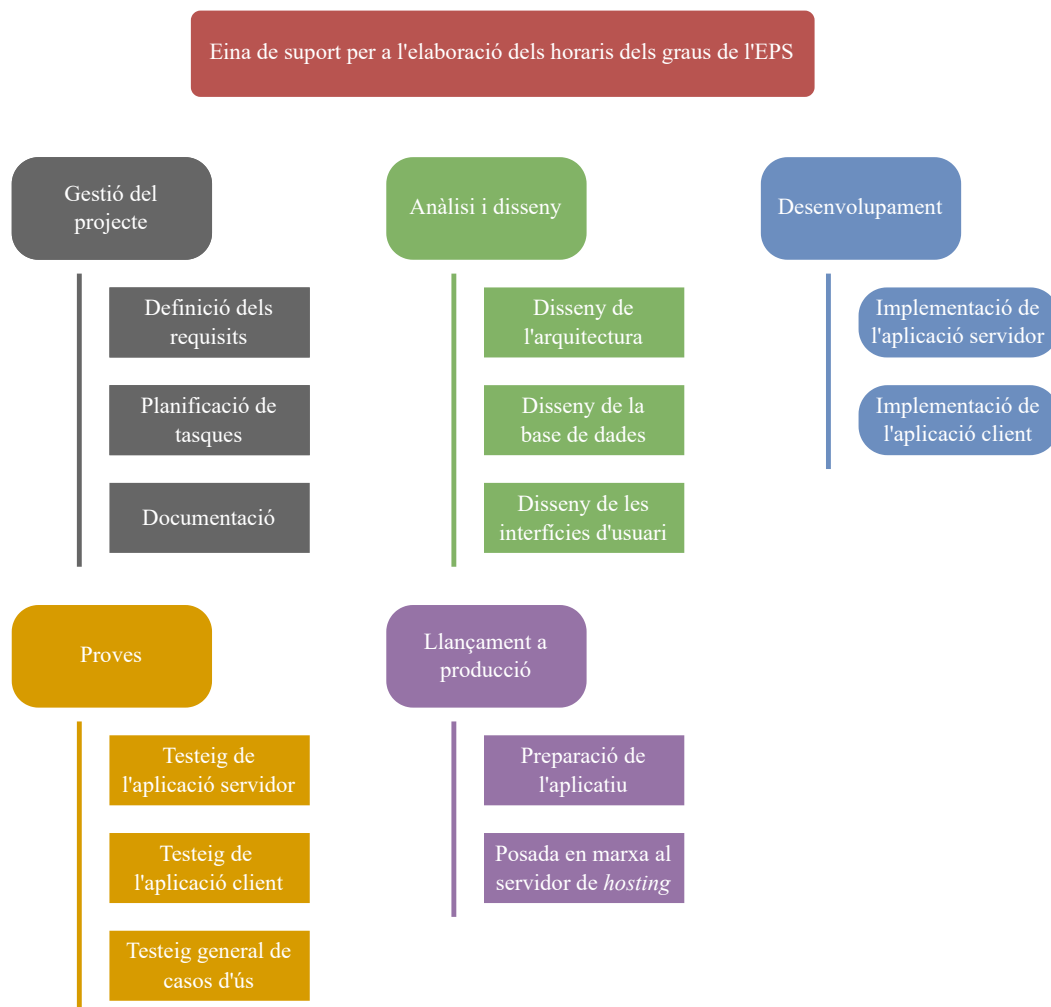


Figura 6.1: Descomposició general de les tasques del projecte.

6.1.3 Gestió del projecte

En aquesta subsecció, s'exposaran els paquets de treball que formen part de la branca de planificació *Gestió del projecte*.

Tal com es pot veure a la figura 6.2, els paquets de treball de la gestió del projecte són la definició dels requisits, la planificació de tasques i la documentació (veure taules 6.1, 6.2 i 6.3, respectivament).

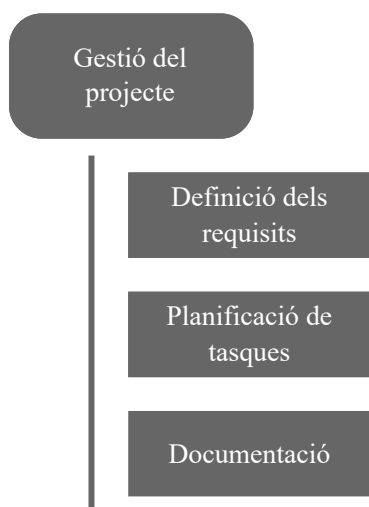


Figura 6.2: Esquema de la branca de planificació *Gestió del projecte*.

PT_1.1: Definició dels requisits	
Descripció	Anàlisi dels requisits que ha de complir l'aplicatiu del projecte.
Tasques	T1: Anàlisi i definició dels requisits no funcionals. T2: Anàlisi i definició dels requisits de domini. T3: Anàlisi i definició dels requisits funcionals. T4: Confecció de la matriu de dependències entre els requisits. T5: Verificació i comentari dels requisits amb els tutors. T6: Correcció del document de requisits.
Temporalització	45 hores.
Lliurable	Document de requisits del sistema (veure capítol 5).

Taula 6.1: Taula del paquet de treball *Definició dels requisits*.

PT_1.2: Planificació de tasques	
Descripció	Planificació de totes les tasques que involucra el projecte.
Tasques	T1: Descomposició i agrupació general de les tasques. T2: Determinació i elaboració dels paquets de treball de cada grup de tasques. T3: Confecció de la matriu de traçabilitat. T4: Estimació del temps necessari per al desenvolupament del projecte i elaboració del cronograma.
Temporalització	30 hores.
Lliurable	Documents de planificació i temporalització del projecte (veure capítol 6).

Taula 6.2: Taula del paquet de treball *Planificació de tasques*.

PT_1.3: Documentació	
Descripció	Confecció del document de la memòria del projecte.
Tasques	T1: Redacció de la memòria del projecte. T2: Verificació i comentari de la memòria amb els tutors. T3: Correcció i retocament final de la memòria.
Temporalització	200 hores.
Lliurable	Memòria del projecte.

Taula 6.3: Taula del paquet de treball *Documentació*.

6.1.4 Anàlisi i disseny

En aquesta subsecció, s'exposaran els paquets de treball que formen part de la branca de planificació *Anàlisi i disseny*.

Tal com es pot veure a la figura 6.3, els paquets de treball de l'anàlisi i disseny són: el disseny de l'arquitectura, el disseny de la base de dades i el disseny de les interfícies d'usuari (veure taules 6.4, 6.5 i 6.6, respectivament).

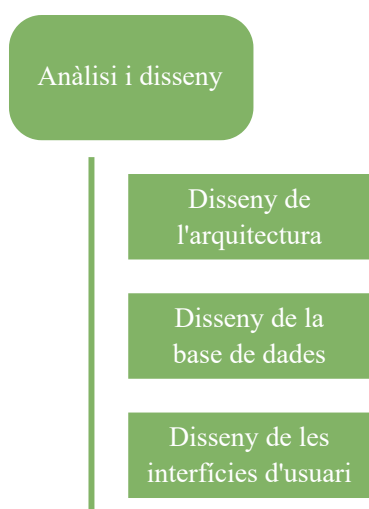


Figura 6.3: Esquema de la branca de planificació *Anàlisi i disseny*.

PT_2.1: Disseny de l'arquitectura	
Descripció	Estudi i disseny de l'arquitectura de l'aplicatiu del projecte.
Tasques	T1: Estudi dels requisits. T2: Estudi i decisió del tipus de base de dades que s'utilitzarà i del seu sistema gestor. T3: Estudi i decisió del tipus de tecnologia amb la qual s'implementarà l'aplicació web client. T4: Estudi i decisió del tipus de tecnologia amb la qual s'implementarà l'aplicació servidor.
Temporalització	15 hores.
Lliurable	Document de valoració i justificació de les tecnologies que conformen l'arquitectura de l'aplicatiu del projecte.

Taula 6.4: Taula del paquet de treball *Disseny de l'arquitectura*.

PT_2.2: Disseny de la base de dades	
Descripció	Disseny de l'estructura de la base de dades de l'aplicatiu i determinació de la informació que s'hi emmagatzemarà.
Tasques	<p>T1: Detall de les entitats que conformaran la base de dades.</p> <p>T2: Decisió dels camps en què es dividiran les entitats, juntament amb el seu tipus i configuració.</p> <p>T3: Determinació de les relacions que existiran entre les diferents entitats.</p> <p>T4: Elaboració de l'esquema del model relacional.</p>
Temporalització	25 hores.
Lliurable	Esquema del model relacional de la base de dades.

Taula 6.5: Taula del paquet de treball *Disseny de la base de dades*.

PT_2.3: Disseny de les interfícies d'usuari	
Descripció	Disseny de les interfícies d'usuari que es presentaran a l'aplicació web client.
Tasques	<p>T1: Confecció dels esquemes de les interfícies d'usuari de les diferents parts de l'aplicació.</p> <p>T2: Anàlisi dels esquemes i verificació de la seva adaptació als requisits.</p> <p>T3: Plantejament de possibles nous requisits que hagin pogut sorgir a partir dels esquemes.</p>
Temporalització	85 hores.
Lliurable	Esquemes de les interfícies d'usuari de l'aplicació web client.

Taula 6.6: Taula del paquet de treball *Disseny de les interfícies d'usuari*.

6.1.5 Desenvolupament

En aquesta subsecció, s'exposaran els paquets de treball que formen part de la branca de planificació *Desenvolupament* (veure figura 6.4).

A diferència de la resta de branques, aquesta no conté directament un conjunt de paquets de treball, sinó que en deriven altres branques. Als apartats següents es detallaran els paquets de treball que conté cadascuna.

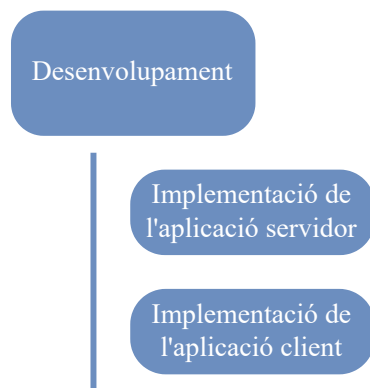


Figura 6.4: Esquema de la branca de planificació *Desenvolupament*.

6.1.5.1 Implementació de l'aplicació servidor

Tal com es pot veure a la figura 6.5, els paquets de treball de la implementació de l'aplicació servidor són: la preparació inicial, la configuració de la base de dades, la definició dels models de dades, la implementació de l'enrutament, la implementació del *login* i del registre i la implementació dels serveis (veure taules 6.7, 6.8, 6.9, 6.10, 6.11 i 6.12, respectivament).

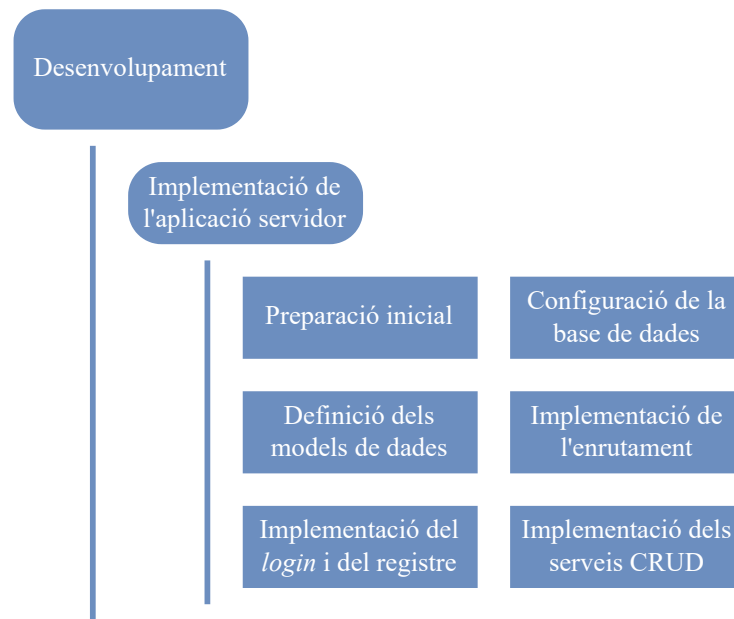


Figura 6.5: Esquema de la branca de planificació derivada *Implementació de l'aplicació servidor*.

PT_3.1.1: Preparació inicial	
Descripció	Preparació i configuració inicial de l'aplicació servidor.
Tasques	T1: Preparació de l'entorn de desenvolupament. T2: Configuració del procés d'escolta de peticions. T3: Implementació del tractament general de peticions i d'errors. T4: Implementació del sistema de depuració i monitoreig. T5: Implementació del sistema de configuració i variables d'entorn.
Temporalització	25 hores.
Lliurable	Aplicació servidor preparada per ser implementada.

Taula 6.7: Taula del paquet de treball *Preparació inicial de l'aplicació servidor*.

PT_3.1.2: Configuració de la base de dades	
Descripció	Configuració inicial del sistema gestor de la base de dades i de l'ORM (<i>object relational mapper</i>).
Tasques	T1: Configuració de la connexió entre l'aplicació de l'API i el sistema gestor de la base de dades. T2: Configuració de l'ORM escollit. T3: Determinació dels usuaris del sistema gestor de la base de dades.
Temporalització	5 hores.
Lliurable	Sistema gestor de la base de dades configurat.

Taula 6.8: Taula del paquet de treball *Configuració de la base de dades*.

PT_3.1.3: Definició dels models de dades	
Descripció	Definició dels models de dades que s'han de guardar a la base de dades.
Tasques	<p>T1: Definició i configuració dels models de dades descrits a l'esquema del model relacional.</p> <p>T2: Implementació de validadors de dades per a cada camp dels models.</p> <p>T3: Implementació i configuració de les relacions entre models.</p>
Temporalització	30 hores.
Lliurable	Models de dades definits.

Taula 6.9: Taula del paquet de treball *Definició dels models de dades*.

PT_3.1.4: Implementació de l'enrutament	
Descripció	Definició dels diferents <i>endpoints</i> de l'API i implementació de l'enrutament de peticions.
Tasques	<p>T1: Definició dels <i>endpoints</i> de l'API i de les seves rutes.</p> <p>T2: Implementació de l'enrutament de peticions a l'<i>endpoint</i> corresponent.</p> <p>T3: Implementació dels mecanismes d'autorització segons el rol de l'usuari.</p>
Temporalització	20 hores.
Lliurable	Enrutament implementat.

Taula 6.10: Taula del paquet de treball *Implementació de l'enrutament*.

PT_3.1.5: Implementació del <i>login</i> i del registre	
Descripció	Implementació del tractament de peticions d'autenticació i de registre d'usuaris.
Tasques	<p>T1: Implementació inicial dels processos d'autenticació i de registre.</p> <p>T2: Implementació dels mecanismes de seguretat involucrats.</p> <p>T3: Implementació de l'enviament dels correus electrònics involucrats.</p>
Temporalització	30 hores.
Lliurable	<i>Login</i> i registre implementats.

Taula 6.11: Taula del paquet de treball *Implementació del login i del registre*.

PT_3.1.6: Implementació dels serveis CRUD.	
Descripció	Implementació dels serveis de creació, recuperació, modificació i eliminació per a cadascun dels models de dades.
Tasques	<p>T1: Implementació dels serveis CRUD per als models de dades relacionats amb els Administradors.</p> <p>T2: Implementació dels serveis CRUD per als models de dades relacionats amb els Coordinadors.</p> <p>T3: Implementació dels serveis CRUD per als models de dades relacionats amb els Directors de departament.</p> <p>T4: Implementació dels serveis CRUD per als models de dades relacionats amb els Responsables de docència.</p> <p>T5: Implementació dels serveis CRUD per als models de dades relacionats amb els Professors.</p>
Temporalització	85 hores.
Lliurable	Serveis de CRUD implementats.

Taula 6.12: Taula del paquet de treball *Implementació dels serveis CRUD*.

6.1.5.2 Implementació de l'aplicació client

Tal com es pot veure a la figura 6.6, els paquets de treball de l'implementació de l'aplicació client són: la preparació inicial de l'aplicació client, la implementació del *login* i relacionats, la implementació de la part dels Administradors, la implementació de la part dels Coordinadors, la implementació de la part dels Directors de departament, la implementació de la part dels Responsables de docència i la implementació de la part dels Professors (veure taules 6.13, 6.14, 6.15, 6.16, 6.17, 6.18 i 6.19, respectivament).

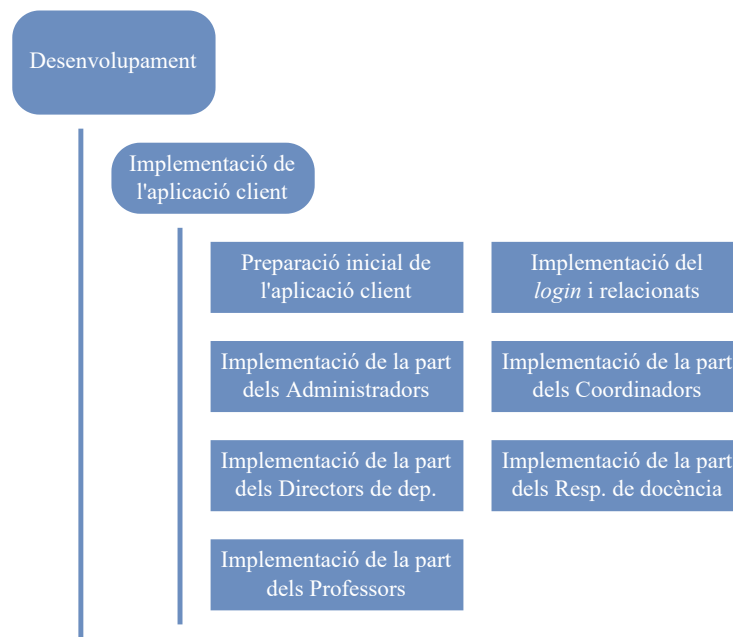


Figura 6.6: Esquema de la branca de planificació derivada *Implementació del client*.

PT_3.2.1: Preparació inicial de l'aplicació client	
Descripció	Preparació i configuració inicials de l'aplicació client.
Tasques	T1: Preparació de l'entorn de desenvolupament.
Temporalització	10 hores.
Lliurable	Aplicació client preparada per ser implementada.

Taula 6.13: Taula del paquet de treball *Preparació inicial de l'aplicació client*.

PT_3.2.2: Implementació del <i>login</i> i relacionats	
Descripció	Implementació de les finestres i procediments relacionats amb l'autenticació dels usuaris.
Tasques	T1: Implementació de la finestra d'autenticació. T2: Implementació del conjunt de finestres de restabliment de contrasenya.
Temporalització	40 hores.
Lliurable	Finestres i procediments de <i>login</i> i relacionats implementats.

Taula 6.14: Taula del paquet de treball *Implementació del login i relacionats*.

PT_3.2.3: Implementació de la part dels Administradors	
Descripció	Implementació de les finestres i procediments que involucren els usuaris amb rol d'Administrador.
Tasques	T1: Implementació del conjunt de finestres relacionades amb la gestió de plans docents. T2: Implementació del conjunt de finestres relacionades amb l'assignació d'aules. T3: Implementació del conjunt de finestres relacionades amb la gestió de Coordinadors. T4: Implementació del conjunt de finestres relacionades amb la gestió de Directors de departament.
Temporalització	80 hores.
Lliurable	Part dels usuaris Administradors implementada.

Taula 6.15: Taula del paquet de treball *Implementació de la part dels Administradors*.

PT_3.2.4: Implementació de la part dels Coordinadors	
Descripció	Implementació de les finestres i procediments que involucren els usuaris amb rol de Coordinador.
Tasques	<p>T1: Implementació del conjunt de finestres relacionades amb la gestió d'horaris d'estudis.</p> <p>T2: Implementació del conjunt de finestres relacionades amb la consulta d'horaris de Professors.</p> <p>T3: Implementació del conjunt de finestres relacionades amb la consulta d'horaris d'aules.</p>
Temporalització	150 hores.
Lliurable	Part dels usuaris Coordinadors implementada.

Taula 6.16: Taula del paquet de treball *Implementació de la part dels Coordinadors*.

PT_3.2.5: Implementació de la part dels Directors de departament	
Descripció	Implementació de les finestres i procediments que involucren els usuaris amb rol de Director de departament.
Tasques	<p>T1: Implementació del conjunt de finestres relacionades amb la consulta d'horaris de Professors.</p> <p>T2: Implementació del conjunt de finestres relacionades amb la consulta d'horaris d'estudis.</p> <p>T3: Implementació del conjunt de finestres relacionades amb la gestió de Responsables de docència.</p>
Temporalització	15 hores.
Lliurable	Part dels usuaris Directors de departament implementada.

Taula 6.17: Taula del paquet de treball *Implementació de la part dels Directors de departament*.

PT_3.2.6: Implementació de la part dels Responsables de docència	
Descripció	Implementació de les finestres i procediments que involucren els usuaris amb rol de Responsable de docència.
Tasques	<p>T1: Implementació del conjunt de finestres relacionades amb la consulta d'horaris de Professors.</p> <p>T2: Implementació del conjunt de finestres relacionades amb l'assignació de Professors.</p> <p>T3: Implementació del conjunt de finestres relacionades amb la gestió de Professors.</p>
Temporalització	40 hores.
Lliurable	Part dels usuaris Responsables de docència implementada.

Taula 6.18: Taula del paquet de treball *Implementació de la part dels Responsables de docència*.

PT_3.2.7: Implementació de la part dels Professors	
Descripció	Implementació de les finestres i procediments que involucren els usuaris amb rol de Professor.
Tasques	<p>T1: Implementació del conjunt de finestres relacionades amb la consulta dels propis horaris.</p> <p>T2: Implementació del conjunt de finestres relacionades amb la consulta d'horaris d'assignatures.</p> <p>T3: Implementació del conjunt de finestres relacionades amb la consulta d'horaris de estudis.</p>
Temporalització	15 hores.
Lliurable	Part dels usuaris Professors implementada.

Taula 6.19: Taula del paquet de treball *Implementació de la part dels Professors*.

6.1.6 Proves

En aquesta subsecció, s'exposaran els paquets de treball que formen part de la branca de planificació *Proves*.

Tal com es pot veure a la figura 6.7, els paquets de treball de les proves són el testeig de l'aplicació servidor, el testeig de l'aplicació client i el testeig general de casos d'ús (veure taules 6.20, 6.21 i 6.22, respectivament).

Cal destacar que el testeig tant de l'aplicació client com de l'aplicació servidor no està previst efectuar-se només al final de tota la fase de desenvolupament, sinó que s'anirà fent de manera progressiva a mesura que avança la implementació.



Figura 6.7: Esquema de la branca de planificació *Proves*.

PT_4.1: Testeig de l'aplicació servidor	
Descripció	Realització de proves manuals i depuració de l'aplicació servidor.
Tasques	T1: Disseny de les proves que s'han de realitzar. T2: Efectuació de les proves i anàlisi dels resultats. T3: Correcció dels errors que hagin pogut aparèixer.
Temporalització	40 hores.
Lliurable	Proves i depuració de l'aplicació servidor realitzades.

Taula 6.20: Taula del paquet de treball *Testeig de l'aplicació servidor*.

PT_4.2: Testeig de l'aplicació client	
Descripció	Realització de proves manuals i depuració de l'aplicació client.
Tasques	T1: Disseny de les proves que s'han de realitzar. T2: Efectuació de les proves i anàlisi dels resultats. T3: Correcció dels errors que hagin pogut aparèixer.
Temporalització	50 hores.
Lliurable	Proves i depuració de l'aplicació client realitzada.

Taula 6.21: Taula del paquet de treball *Testeig de l'aplicació client*.

PT_4.3: Testeig general de casos d'ús	
Descripció	Realització de proves i depuració replicant manualment casos d'ús reals.
Tasques	T1: Simulació de situacions reals a mode de prova. T2: Correcció dels errors que hagin pogut aparèixer.
Temporalització	20 hores.
Lliurable	Proves i depuració en un entorn pre-productiu realitzada.

Taula 6.22: Taula del paquet de treball *Testeig general de casos d'ús*.

6.1.7 Llançament a producció

En aquesta subsecció, s'exposaran els paquets de treball que formen part de la branca de planificació *Llançament a producció*.

Tal com es pot veure a la figura 6.8, els paquets de treball del llançament a producció són la preparació de l'aplicatiu i la posada en marxa al servidor de *hosting* (veure taules 6.23 i 6.24, respectivament).

Cal destacar que la posada en marxa en l'entorn productiu no forma part l'abast del projecte i que, per tant, no es comptabilitzarà en el còmput de temps total. Així i tot, per tal d'obtenir una planificació completa, s'ha decidit contemplar-ho.

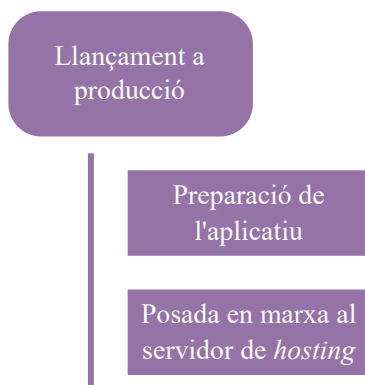


Figura 6.8: Esquema de la branca de planificació *Llançament a producció*.

PT_5.1: Preparació de l'aplicatiu	
Descripció	Preparació de l'aplicatiu per al llançament a producció.
Tasques	T1: Revisió i neteja del codi font. T2: Verificació que no es mostra cap informació en pantalla o en consola que pugui comprometre la seguretat. T3: Comprovació que les variables d'entorn es recuperen correctament. T4: Verificar els arxius de configuració.
Temporalització	10 hores.
Lliurable	Aplicatiu preparat per al llançament a producció.

Taula 6.23: Taula del paquet de treball *Preparació de l'aplicatiu*.

PT_5.2: Posada en marxa al servidor de <i>hosting</i>	
Descripció	Llançament a producció de l'aplicatiu al servidor de <i>hosting</i> .
Tasques	T1: Configuració general del servidor. T2: Traspàs de l'aplicatiu al servidor. T3: Establiment de les variables d'entorn corresponents. T4: Posada en marxa de l'aplicatiu i realització de comprovacions.
Temporalització	25 hores.
Lliurable	Aplicatiu en funcionament a l'entorn productiu.

Taula 6.24: Taula del paquet de treball *Posada en marxa al servidor de hosting*.

6.2 Matriu de traçabilitat

En aquesta secció, es representarà la matriu de traçabilitat que relaciona els paquets de treball definits a la secció 6.1 i els requisits definits al capítol 5.

D'aquesta manera, es podrà saber quins requisits s'assoleixen un cop completades les tasques dels paquets de treball.

El format que adoptarà cadascuna de les relacions es defineix a continuació:

$$(RF-X, RF-Y, \dots) \longrightarrow (PT-P, PT-Q, \dots)$$

És a dir, la llista de requisits de la part anterior de la flexa s'assoleix un cop completada la de paquets de treball situada a la part posterior.

La matriu de traçabilitat completa és la següent:

- (RF-G1, RF-G3) \longrightarrow (PT_3.1.5, PT_3.2.2)
- (RF-G2) \longrightarrow (PT_3.1.5, PT_3.2.3, PT_3.2.5, PT_3.2.6)
- (RF-G5, RF-G6) \longrightarrow (PT_3.1.6, PT_3.2.3, PT_3.2.4, PT_3.2.5, PT_3.2.6, PT_3.2.7)
- (RF-G4) \longrightarrow (PT_3.1.4)
- (RF-G7) \longrightarrow (PT_2.2, PT_3.1.2, PT_3.1.3)
- (RF-A1, RF-A2, RF-A3, RF-A4, RF-A5, RF-A6, RF-A7, RF-A8, RF-A9, RF-A11, RF-A12, RF-A13, RF-A14, RF-A15, RF-A16, RF-A18) \longrightarrow (PT_3.1.6, PT_3.2.3)
- (RF-A10, RF-A17) \longrightarrow (PT_3.1.5, PT_3.1.6, PT_3.2.3)
- (RF-C1, RF-C2, RF-C3, RF-C4, RF-C5) \longrightarrow (PT_3.1.6, PT_3.2.4)
- (RF-D1, RF-D2, RF-D3, RF-D4, RF-D5, RF-D6, RF-D7, RF-D9) \longrightarrow (PT_3.1.6, PT_3.2.5)
- (RF-D8) \longrightarrow (PT_3.1.5, PT_3.1.6, PT_3.2.5)
- (RF-R1, RF-R2, RF-R3, RF-R4, RF-R6) \longrightarrow (PT_3.1.6, PT_3.2.6)
- (RF-R5, RF-G3) \longrightarrow (PT_3.1.5, PT_3.1.6, PT_3.2.6)
- (RF-P1, RF-P2, RF-P3) \longrightarrow (PT_3.1.6, PT_3.2.7)

6.3 Temporalització

6.3.1 Determinació d'activitats

En aquesta subsecció, a partir de la matriu de dependències (veure secció 5.5) i la de traçabilitat (veure secció 6.2), es determinaran les activitats que caldrà dur a terme per completar el desenvolupament del projecte.

Una activitat és una agrupació de paquets de treball i/o de tasques de paquets de treball que cal completar per poder assolir una sèrie de requisits.

Un cop s'hagin determinat les activitats, s'efectuarà una estimació del temps que podria requerir completar cadascuna. Aquesta estimació consistirà, primerament, en indicar tres valors de temps:

- $t(O)$: Estimació optimista.
- $t(M)$: El més probable.
- $t(P)$: Estimació pessimista.

Tot seguit, es calcularà el temps esperat de duració $t(E)$ de l'activitat. Aquest càlcul consisteix en una mitjana ponderada dels valors de l'estimació anterior, tal com es pot veure a continuació:

$$t(E) = \frac{t(O) + 4t(M) + t(P)}{6}$$

Les taules marcades en gris (6.25, 6.26 i 6.45) corresponen a la fase de gestió del projecte, les marcades en verd (6.27, 6.28 i 6.29) a la d'anàlisi i disseny, les marcades en blau (6.30, 6.31, 6.32, 6.33, 6.34, 6.35, 6.36, 6.37, 6.38, 6.39 i 6.40) a la de desenvolupament, les marcades en taronja (6.41, 6.42 i 6.43) a la de proves i, finalment, la marcada en lila (6.44) a la de llançament a producció

A1: Definició de requisits	
Tasques	PT_1.1.
Estimació de temps	$t(O)$: 30 hores. $t(M)$: 45 hores. $t(P)$: 60 hores.
Durada esperada	$t(E)$: 45 hores.

Taula 6.25: Taula de l'activitat *Definició de requisits*.

A2: Planificació de tasques	
Tasques	PT_1.2.
Estimació de temps	$t(O)$: 25 hores. $t(M)$: 30 hores. $t(P)$: 50 hores.
Durada esperada	$t(E)$: 33 hores.

Taula 6.26: Taula de l'activitat *Planificació de tasques*.

A3: Disseny de l'arquitectura	
Tasques	PT_2.1.
Estimació de temps	$t(O)$: 10 hores. $t(M)$: 15 hores. $t(P)$: 30 hores.
Durada esperada	$t(E)$: 17 hores.

Taula 6.27: Taula de l'activitat *Disseny de l'arquitectura*.

A4: Disseny de la base de dades	
Tasques	PT_2.2.
Estimació de temps	$t(O)$: 15 hores. $t(M)$: 25 hores. $t(P)$: 30 hores.
Durada esperada	$t(E)$: 24 hores.

Taula 6.28: Taula de l'activitat *Disseny de la base de dades*.

A5: Disseny de les interfícies d'usuari	
Tasques	PT_2.3.
Estimació de temps	$t(O)$: 60 hores. $t(M)$: 85 hores. $t(P)$: 120 hores.
Durada esperada	$t(E)$: 87 hores.

Taula 6.29: Taula de l'activitat *Disseny de les interfícies d'usuari*.

A6: Preparació inicial de l'aplicació servidor	
Tasques	PT_3.1.1.
Estimació de temps	$t(O)$: 20 hores. $t(M)$: 25 hores. $t(P)$: 30 hores.
Durada esperada	$t(E)$: 25 hores.

Taula 6.30: Taula de l'activitat *Preparació inicial de l'aplicació servidor*.

A7: Configuració de la base de dades	
Tasques	PT_3.1.2.
Estimació de temps	$t(O)$: 4 hores. $t(M)$: 5 hores. $t(P)$: 20 hores.
Durada esperada	$t(E)$: 8 hores.

Taula 6.31: Taula de l'activitat *Configuració de la base de dades*.

A8: Definició dels models de dades	
Tasques	PT_3.1.3.
Estimació de temps	$t(O)$: 25 hores. $t(M)$: 30 hores. $t(P)$: 35 hores.
Durada esperada	$t(E)$: 30 hores.

Taula 6.32: Taula de l'activitat *Definició dels models de dades*.

A9: Implementació de l'enrutament	
Tasques	PT_3.1.4.
Estimació de temps	$t(O)$: 15 hores. $t(M)$: 20 hores. $t(P)$: 25 hores.
Durada esperada	$t(E)$: 25 hores.

Taula 6.33: Taula de l'activitat *Implementació de l'enrutament*.

A10: Preparació inicial de l'aplicació client	
Tasques	PT_3.2.1.
Estimació de temps	$t(O)$: 8 hores. $t(M)$: 10 hores. $t(P)$: 15 hores.
Durada esperada	$t(E)$: 11 hores.

Taula 6.34: Taula de l'activitat *Preparació inicial de l'aplicació client*.

A11: Implementació del login i relacionats	
Tasques	PT_3.1.5, PT_3.2.2.
Estimació de temps	$t(O)$: 55 hores. $t(M)$: 70 hores. $t(P)$: 80 hores.
Durada esperada	$t(E)$: 69 hores.

Taula 6.35: Taula de l'activitat *Implementació del login i relacionats*.

A12: Implementació de la part dels Administradors	
Tasques	[T1] de PT_3.1.6, PT_3.2.3.
Estimació de temps	$t(O)$: 90 hores. $t(M)$: 95 hores. $t(P)$: 110 hores.
Durada esperada	$t(E)$: 97 hores.

Taula 6.36: Taula de l'activitat *Implementació de la part dels Administradors*.

A13: Implementació de la part dels Coordinadors	
Tasques	[T2] de PT_3.1.6, PT_3.2.4.
Estimació de temps	$t(O)$: 140 hores. $t(M)$: 165 hores. $t(P)$: 180 hores.
Durada esperada	$t(E)$: 164 hores.

Taula 6.37: Taula de l'activitat *Implementació de la part dels Coordinadors*.

A14: Implementació de la part dels Directors de departament	
Tasques	[T3] de PT_3.1.6, PT_3.2.5.
Estimació de temps	$t(O)$: 20 hores. $t(M)$: 30 hores. $t(P)$: 40 hores.
Durada esperada	$t(E)$: 30 hores.

Taula 6.38: Taula de l'activitat *Implementació de la part dels Directors de departament*.

A15: Implementació de la part dels Responsables de docència	
Tasques	[T4] de PT_3.1.6, PT_3.2.6.
Estimació de temps	$t(O)$: 50 hores. $t(M)$: 55 hores. $t(P)$: 70 hores.
Durada esperada	$t(E)$: 57 hores.

Taula 6.39: Taula de l'activitat *Implementació de la part dels Responsables de docència*.

A16: Implementació de la part dels Professors	
Tasques	[T5] de PT_3.1.6, PT_3.2.7.
Estimació de temps	$t(O)$: 10 hores. $t(M)$: 30 hores. $t(P)$: 40 hores.
Durada esperada	$t(E)$: 30 hores.

Taula 6.40: Taula de l'activitat *Implementació de la part dels Professors*.

A17: Testeig de l'aplicació servidor	
Tasques	PT_4.1.
Estimació de temps	$t(O)$: 20 hores. $t(M)$: 40 hores. $t(P)$: 50 hores.
Durada esperada	$t(E)$: 39 hores.

Taula 6.41: Taula de l'activitat *Testeig de l'aplicació servidor*.

A18: Testeig de l'aplicació client	
Tasques	PT_4.2.
Estimació de temps	$t(O)$: 30 hores. $t(M)$: 50 hores. $t(P)$: 60 hores.
Durada esperada	$t(E)$: 49 hores.

Taula 6.42: Taula de l'activitat *Testeig de l'aplicació client*.

A19: Testeig general de casos d'ús	
Tasques	PT_4.3.
Estimació de temps	$t(O)$: 10 hores. $t(M)$: 20 hores. $t(P)$: 30 hores.
Durada esperada	$t(E)$: 20 hores.

Taula 6.43: Taula de l'activitat *Testeig general de casos d'ús*.

A20: Posada en marxa de l'aplicatiu	
Tasques	PT_5.1, PT_5.2.
Estimació de temps	$t(O)$: 25 hores. $t(M)$: 35 hores. $t(P)$: 50 hores.
Durada esperada	$t(E)$: 36 hores.

Taula 6.44: Taula de l'activitat *Posada en marxa de l'aplicatiu*.

A21: Documentació	
Tasques	PT_1.3.
Estimació de temps	$t(O)$: 150 hores. $t(M)$: 200 hores. $t(P)$: 220 hores.
Durada esperada	$t(E)$: 195 hores.

Taula 6.45: Taula de l'activitat *Documentació*.

6.3.2 Diagrama d'activitats

En aquesta subsecció, s'il·lustrarà el diagrama de les activitats planificades a la subsecció 6.3.1.

Tal com es pot veure a la figura 6.9, el diagrama consistirà en un graf els vèrtexs del qual representen activitats i les arestes relacions “acabar per començar”. Les arestes també duran el temps esperat de durada de l'activitat que tenen com a origen.

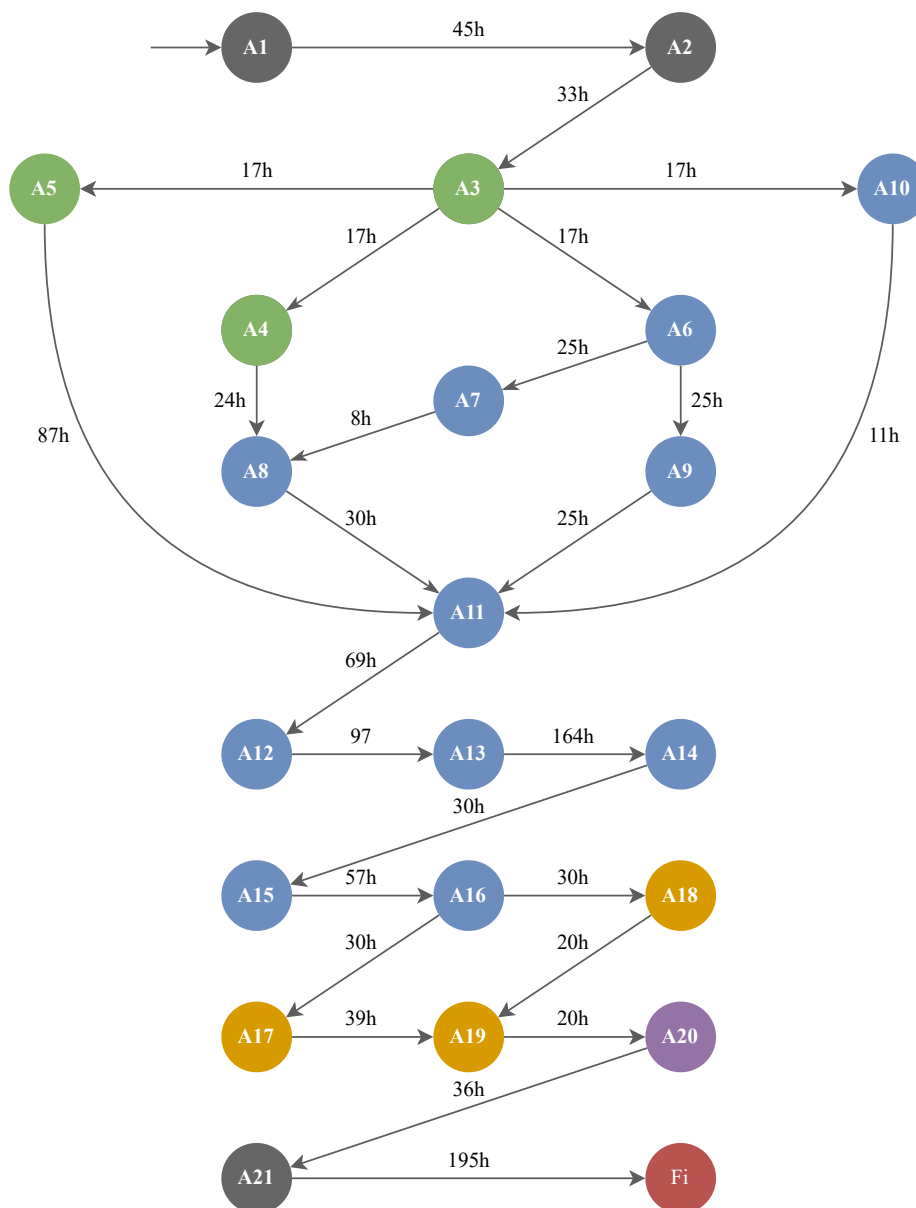


Figura 6.9: Diagrama de les activitats planificades.

A partir del diagrama d'activitats, es pot saber fàcilment la durada màxima aproximada del projecte. Per calcular-la, cal sumar els temps de cadascuna de les activitats del camí crític del graf, és a dir, el camí des d'A1 fins a Fi més costós en temps.

A1 → A2 → A3 → A5 → A11 → A12 → A13 → A14 → A15 → A16 → A17 → A19 → A20 → A21 → Fi

El resultat de la suma dels temps del camí crític i, per tant, la durada màxima estimada del projecte seria de 864 hores (669h de planificació i desenvolupament més 195h de documentació). No obstant això, aquest càlcul s'ha realitzat sense tenir en compte que el projecte es durà a terme de manera individual, ja que hi ha activitats que es poden realitzar paral·lelament.

Per tant, si es considera que les activitats no es paral·lelitzzen, la durada màxima estimada del projecte és de **1055 hores** (860h de planificació i desenvolupament més 195h de documentació).

6.3.3 Cronograma

En aquesta subsecció, es representarà el diagrama de Gantt que recull la planificació del projecte a partir de les activitats determinades a la subsecció 6.3.1 i n'estima les dates (veure figura 6.10).

Com es pot apreciar en el cronograma, el temps necessari per desenvolupar la totalitat del projecte excedeix el temps de què es disposa. A causa d'això, se n'hauran d'acotar lleugerament els objectius.

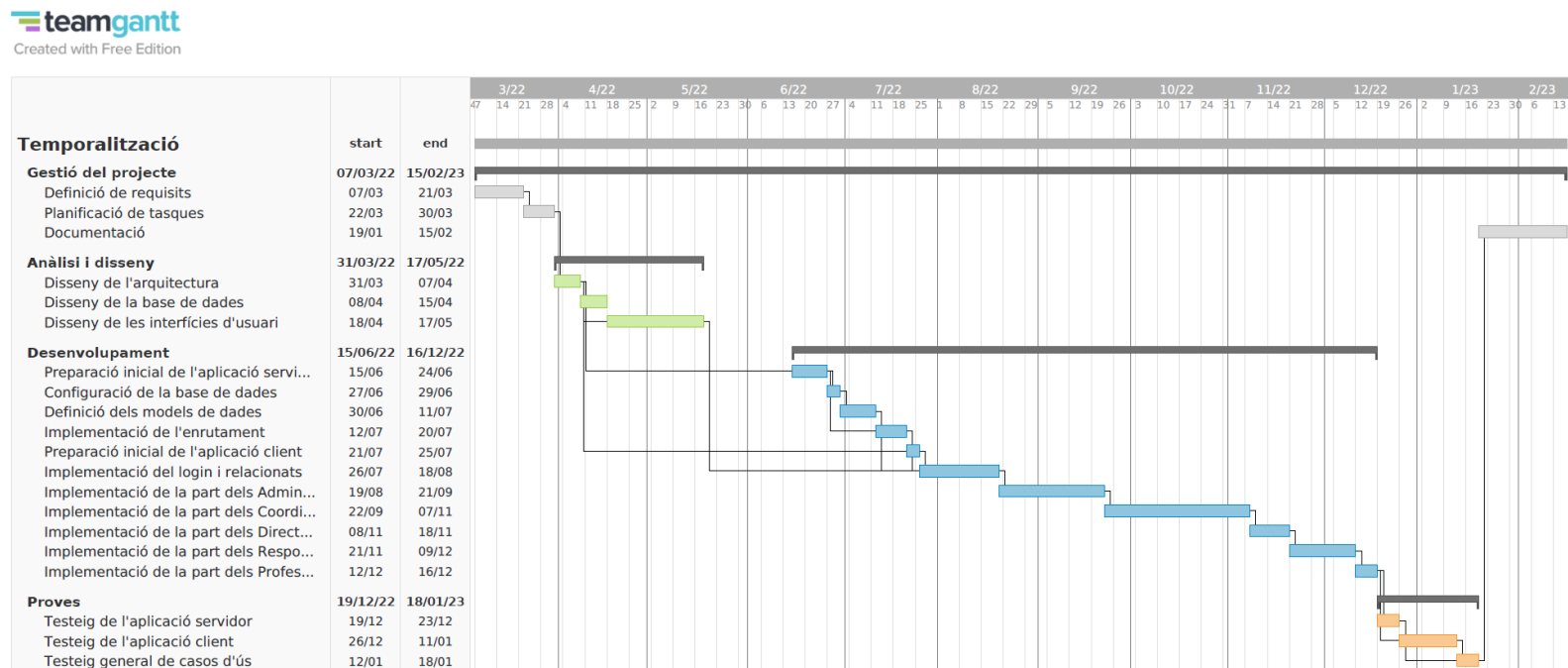


Figura 6.10: Diagrama de Gantt.

Estudis i decisions

En aquest capítol, primerament, es parlarà sobre com s'ha gestionat i estructurat el projecte en quant al codi (veure secció 7.1). Tot seguit, ja s'entrarà en detall en les tecnologies utilitzades pel desenvolupament del programari, tant de l'aplicació client com de l'aplicació servidor (veure seccions 7.2 i 7.3, respectivament). També s'explicaran les decisions que involucren la base de dades (veure secció 7.4). En darrer lloc, s'indicaran les eines que s'han fet servir tant per al desenvolupament del programari com per a la confecció de la memòria (veure seccions 7.5 i 7.6, respectivament)

7.1 Gestió i estructuració del projecte

7.1.1 Control de versions

En aquesta subsecció, es justificarà l'elecció del sistema de control de versions utilitzat en el projecte: **Git**.

Disposar d'un sistema de control de versions a l'hora de desenvolupar una aplicació aporta un gran nombre d'avantatges, d'entre els quals destaca la capacitat de consultar o restaurar estats antics del codi i la de bifurcar el desenvolupament.

Git [Scott Chacon, Jason Long, *contribuïdors de Git 2022*] (veure figura 7.1), és el sistema de control de versions més popular amb diferència i, encara que existeixin algunes alternatives, és la que més funcionalitats ofereix i l'opció preferida de la gran majoria de desenvolupadors. A banda d'aquests motius, Git és l'única eina d'aquesta tipologia amb què tinc experiència.



Figura 7.1: Logotip de Git.

Normalment, l'ús de **Git** sol acompanyar-se de l'allotjament dels repositoris a una plataforma web dedicada a aquesta funció. Aquests tipus de plataforma

permeten evitar la pèrdua accidental del codi d'un projecte, a banda d'aportar incomptables funcionalitats per al treball col·laboratiu, entre altres.

Hi ha diverses alternatives pel que fa a aquest tipus de plataforma, com ara **GitHub**, GitLab o BitBucket. No obstant això, l'opció escollida per al projecte és **GitHub** [GitHub 2022] (veure figura 7.2), ja que la domino més que la resta i em sento més còmode treballant-hi.



Figura 7.2: Logotip de GitHub.

Per acabar, cal esmentar que aquestes eines s'han fet servir tant pel programari com per a la memòria.

7.1.2 Estructuració del projecte

En aquesta subsecció, es justificarà l'elecció de l'estructura del sistema: **monorepositori**.

Tal com s'ha vist al capítol 4, el projecte engloba el desenvolupament de dues aplicacions: el client i el servidor. Aquestes aplicacions es poden estructurar de dues maneres diferents:

- **Multirepositori:** Separar les dues aplicacions en repositoris diferents.
- **Monorepositori:** Agrupar les dues aplicacions en el mateix repositori.

Els beneficis principals del multirepositori són que els repositoris individuals són més fàcils de gestionar i queden més nets i recollits. En canvi, encara que en el **monorepositori** passi el contrari, és un aspecte que no té gaire pes, ja que l'escala del projecte és petita. A més a més, aquesta alternativa ofereix avantatges més interessants, com ara una millor estandarització del codi i la possibilitat de la seva reutilització, entre altres.

A més a més, tal com es veurà a les seccions 7.2 i 7.3, el llenguatge de programació amb què s'han desenvolupat les dues aplicacions és el mateix: JavaScript. Gràcies a això es pot utilitzar un mateix gestor de paquets per a les dues.

Els gestors de paquets de JavaScript actuals més populars també incorporen altres eines i utilitats que faciliten la gestió d'un **monorepositori** dividint-lo en diferents espais de treball. Permeten definir les dependències de cada espai de treball i també les comunes. Aquest fet fa que la instal·lació i actualització dels paquets sigui més ràpida i eficient que en el cas de l'estructuració del projecte

en multirepositori. Per aquests motius i per comoditat, l'elecció per al projecte ha estat l'estructuració en **monorepositori**.

Pel que fa al gestor de paquets, s'ha optat per **npm** [npm 2022] (veure figura 7.3). Els motius d'aquesta elecció han estat, principalment, el coneixement previ de l'eina i que és el gestor de paquets estàndard de Node.js, l'entorn escollit per a l'aplicació servidor (veure secció 7.3).



Figura 7.3: Logotip de npm.

7.2 Aplicació client

7.2.1 Bundler

En aquesta subsecció, es justificarà l'elecció del *bundler* utilitzat en l'aplicació client: **Vite**.

Tal com s'ha vist al capítol 4, existeixen múltiples eines d'aquesta tipologia que empaqueten el codi de tots els fitxers o mòduls JavaScript per tal de poder ser enviats al navegador en una sola petició, com ara **Vite**, WebPack, Rollup o Parcel.

No obstant això, el codi font de les aplicacions *front-end* modernes pot arribar a ser enorme a causa de l'ús de *frameworks* i biblioteques. Aquest fet comporta, entre altres inconvenients, una mala experiència de desenvolupament, ja que, per exemple, si es fa servir HMR (*hot module replacement*), les modificacions d'arxius poden tardar uns quants segons en veure's reflectides al navegador. A més a més, el temps d'espera necessari per posar en marxa un servidor de desenvolupament pot resultar excessivament llarg.

Gràcies als avenços en l'ecosistema, com ara la disponibilitat de mòduls ES al navegador de forma nativa, han sorgit noves eines que se n'aprofiten i corregeixen els problemes que presenten els *bundlers* comentats anteriorment.

Degut a aquests motius i a que la documentació oficial del *framework* escollit pel *front-end* del projecte (veure subsecció 7.2.2) en recomana l'ús, s'ha triat **Vite**.

Vite [Evan You 2022a] (veure figura 7.4) és una eina de compilació l'objectiu de la qual és proporcionar una experiència de desenvolupament àgil per a projectes web moderns. entre altres funcionalitats, incorpora un servidor de desenvolupament que permet, per exemple, un HMR extremadament ràpid.



Figura 7.4: Logotip de Vite.

7.2.2 Framework

En aquesta subsecció, es justificarà l'elecció del *framework* JavaScript utilitzat en l'aplicació client: **Vue**.

Tal com s'ha vist al capítol 4, els *frameworks* més populars per a *front-end* són React, Angular i **Vue**. Normalment, les tecnologies més populars són les que tenen una comunitat més gran i, per tant, n'existeix més documentació, més articles i més preguntes i respostes sobre quasi bé qualsevol tipus de problema o dubte. Per aquest motiu, d'entrada, l'elecció s'ha disputat entre aquests tres *frameworks*.

No obstant això, Angular [Google 2022a] ha estat la primera opció descartada. En comparació amb les altres alternatives, és la més pesada, més complexa i amb la corva d'aprenentatge més pronunciada, ja que disposa d'una gran quantitat d'eines i possibilitats que, en moltes ocasions, poden resultar excessives i agobiants a l'hora de desenvolupar aplicacions a petita o mitjana escala.

En aquest punt, s'ha realitzat una comparació entre React [Meta Platforms 2022] i **Vue** [Evan You 2022b] tenint en compte els avantatges i inconvenients principals de cadascun [Jeel Patel 2022].

- Avantatges principals de React:
 - Disposa d'una sintaxi pròpia que permet implementar més funcionalitat amb menys codi.
 - No és difícil d'aprendre per algú experimentat en JavaScript.
 - S'ha desenvolupat un conjunt d'eines i utilitats que, en resum, faciliten molt la comprensió i depuració del codi.
- Inconvenients principals de React:
 - La documentació és bastant pobre, ja que el *software* evoluciona a un ritme molt ràpid i no es dóna l'abast a l'hora de documentar-lo.
 - Depèn molt d'altres eines que completen determinades funcionalitats que li falten.

- Degut a la ràpida evolució i actualització, molts desenvolupadors no es senten còmodes en haver d'estar aprenent continuament com es fan determinades coses que, cada poc temps, canvien de manera de fer.
- Avantatges principals de **Vue**:
 - És molt còmode de treballar-hi gràcies a la llegibilitat i simplicitat del codi.
 - És molt lleuger, ja que només pesa 20KB.
 - La documentació és molt completa i detallada, fet que el fa ràpid i senzill d'aprendre per a desenvolupadors sense gaire experiència en aquest món.
 - És capaç de detectar els components que contenen errors.
- Inconvenients principals de **Vue**:
 - És extremadament flexible, fet que facilita l'aparició d'irregularitats i errors.
 - No disposa de gaires eines i utilitats comunes que ajuden a simplificar el desenvolupament.

A l'hora de dur a terme la decisió final, s'ha valorat molt més l'opció més lleugera i que ofereix el codi més simple i llegible, la millor documentació i la corba d'aprenentatge menys pronunciada, ja que no tenia experiència prèvia en el desenvolupament d'aplicacions web. Per tant, **Vue** és el *framework* escollit per a l'aplicació client del projecte.

Vue (veure figura 7.5) és un *framework* JavaScript per desenvolupar interfícies d'usuari. Està construït sobre HTML, CSS i JavaScript i proporciona un model de programació declaratiu basat en components.



Figura 7.5: Logotip de Vue.

El més comú és definir cada component en un fitxer diferent. Aquests fitxers utilitzen l'extensió “.vue” i se'ls anomena SFC (*Single-File Component*). Cadascun està format per les tres parts principals següents:

- *Script*: Conté el codi JavaScript del component.
- *Template*: Conté l'estructura HTML del component.
- *Style*: Conté els estils CSS del component.

7.2.3 Enrutament

En aquesta subsecció, es presentarà l'eina utilitzada per l'enrutament de l'aplicació client: **Vue Router**.

Vue Router [Evan You, Eduardo San Martin Morote 2022] és l'enrutador oficial de Vue i està profundament integrat amb el seu nucli. Gràcies a aquesta utilitat, es poden implementar rutes dinàmiques, rutes basades en components, enrutament dinàmic, historial de rutes, etc.

7.2.4 Gestió d'estats

En aquesta subsecció, es presentarà l'eina utilitzada per la gestió d'estats de l'aplicació client: **Pinia**.

Pinia [Eduardo San Martin Morote 2022] (veure figura 7.6) és l'eina de gestió d'estats oficial de Vue i permet emmagatzemar i manipular dades reactives més enllà de la jerarquia de components i de manera global. A més a més, accepta HMR i disposa d'integracions amb les eines per a desenvolupadors dels navegadors.



Figura 7.6: Logotip de Pinia.

7.2.5 Estètica i funcionalitats addicionals

En aquesta subsecció, es presentaran les eines utilitzades en quant a estètica i en quant a l'aportació de funcionalitats addicionals que simplifiquen el desenvolupament de l'aplicació client i la fan més robusta: **Material Colors, Material Icons i Quasar**.

Material és un sistema de disseny que defineix unes guies d'estil enfocades al desenvolupament d'interfícies d'usuari d'alta qualitat. Per aquest motiu, abans

de crear paletes de colors pròpies i buscar icones variades per al *front-end* del projecte, es va optar per utilitzar **Material Colors** [Google 2022c] per escollir-ne les paletes de colors i **Material Icons** [Google 2022d] per escollir-ne les icones.

L'abast de l'aplicació client és relativament gran i construir-la des de zero, tot i utilitzar un *framework*, hagués suposat un cost temporal massa elevat. Per tal d'agilitzar-ne el desenvolupament, es va decidir incorporar al projecte alguna biblioteca que disposés de components reutilitzables útils i que seguïssin unes certes guies d'estil, com ara botons, seleccionables, etc.

L'alternativa més popular i més completa és Vuetify. No obstant això, aquesta eina encara no suporta l'última versió de Vue (Vue 3) i ha estat descartada. La segona millor opció després de Vuetify és **Quasar** i, encara que no sigui tant completa, ofereix components i funcionalitats molt útils i interessants.

Quasar [PULSARDEV SRL 2022] (veure figura 7.7) és un *framework* de codi obert basat en Vue. Encara que normalment s'utilitzi com a *framework* "principal" d'un projecte, també es pot incorporar a un projecte existent creat amb Vite en forma de *plugin*, com el cas de l'aplicació client d'aquest projecte.



Figura 7.7: Logotip de Quasar.

A continuació s'enumeren algunes de les funcionalitats que incorpora:

- Components reutilitzables que van des de botons fins a taules complexes, arbres, etc.
- Classes CSS globals i possibilitat de definició de variables CSS.
- Eines (*directives*) de Vue que permeten controlar accions de ratolí, entre altres utilitats.
- Eines (*composables*) de Vue que faciliten la separació i reutilització de diàlegs, entre altres utilitats.
- Suport per biblioteques d'icones com, per exemple, **Material Icons**.

7.2.6 Proves

En aquesta subsecció, es presentaran les eines utilitzades per a la realització de proves a l'aplicació client: **Chrome DevTools amb una extensió de Vue**.

Chrome DevTools [Google 2022b] (veure figura 7.8) és un conjunt d'eines per a desenvolupadors que proporciona el navegador Google Chrome. Aquestes eines resulten molt útils a l'hora de comprovar que el *software* funcioni i depurar-lo en cas que es produeixi algun error.



Figura 7.8: Logotip de Chrome DevTools.

A més a més, Vue disposa d'una extensió que s'afegeix a aquestes eines i fa possible la realització de proves i depuració molt més específiques.

7.3 Aplicació servidor

7.3.1 Entorn i llenguatge de programació

En aquesta subsecció, es justificarà l'elecció tant de l'entorn com del llenguatge de programació utilitzats en l'aplicació servidor: **JavaScript en l'entorn de Node.js**.

Tal com s'ha vist al capítol 4, existeix una quantitat considerable d'entorns i llenguatges de programació que permeten desenvolupar aplicacions de *back-end*. D'entrada i per raons de temps, es descarten els llenguatges amb els quals no tinc cap tipus d'experiència.

En aquest punt, les dues alternatives són **JavaScript en l'entorn de Node.js** i Java. Les aplicacions servidor desenvolupades amb Java són una bona opció però, en notables ocasions, els seus *frameworks* comporten una configuració excessiva que requereix bastant temps i pot dur a errors. D'altra banda, els *frameworks JavaScript* que funcionen sobre **Node.js** solen ser molt més lleugers i senzills de fer servir a nivell bàsic. A més a més, **JavaScript** és el llenguatge que més m'interessa aprendre actualment.

Per aquests motius i per evitar la incomoditat d'haver de desenvolupar a la vegada dues aplicacions amb llenguatges diferents, l'alternativa escollida ha estat **JavaScript en l'entorn de Node.js**.

Node.js [OpenJs Foundation 2022b] (veure figura 7.9) és un entorn d'execució de **JavaScript** multiplataforma i de codi obert que funciona fora del navegador i està orientat al processament d'esdeveniments asíncrons. Està dissenyat principalment per crear aplicacions de xarxa escalables. L'escalabilitat és possible gràcies a la seva capacitat d'atendre múltiples connexions simultàniament sense cap possibilitat de bloqueig: utilitza un sol fil d'execució, el qual executa una funció (*callback*) per a cada connexió. Si aquesta funció realitza una operació asíncrona, com ara comunicar-se amb un sistema gestor de bases de dades, **Node.js** no es bloqueja mentre espera que finalitzi l'operació, sinó que n'executa altres que tingui pendents, com per exemple atendre una altra connexió. No obstant això, un cop hagi acabat l'operació, continuarà amb l'execució de la funció.



Figura 7.9: Logotip de Node.js.

7.3.2 Framework

En aquesta subsecció, es presentarà el *framework* JavaScript utilitzat en l'aplicació servidor: **Express.js**.

Express.js [OpenJs Foundation 2022a] (veure figura 7.10) és el *framework* JavaScript que funciona sobre Node.js més popular. Destaca per ser molt lleuger, minimalista i flexible, a més de proporcionar una metodologia ràpida per crear API REST.



Figura 7.10: Logotip d'Express.js.

7.3.3 ORM

En aquesta subsecció, es presentarà l'ORM utilitzat en l'aplicació servidor: **Sequelize**.

Un ORM (*object relational mapper*) és una tècnica o patró arquitectònic que permet comunicar l'aplicació servidor amb un sistema gestor de bases de dades. La seva finalitat és crear models virtuals per a cada taula de la base de dades i generar comandes SQL automàticament a partir d'instruccions executades amb el llenguatge de l'aplicació.

Sequelize [Contribuïdors de Sequelize 2022] (veure figura 7.11) és un ORM desenvolupat per a Node.js que suporta treballar amb diversos SGBD: PostgreSQL, MySQL, SQLite i MSSQL. Ofereix un ampli conjunt de funcionalitats, les principals del qual són: fort suport de transaccions, relacions, múltiples maneres de carregar dades, esborrat no definitiu, etc.



Figura 7.11: Logotip de Sequelize.

7.3.4 Proves

En aquesta subsecció, es justificarà l'elecció de l'eina utilitzada per a la realització de proves a l'API que exposa l'aplicació servidor: **Postman**.

Les dues alternatives que possibiliten realitzar aquestes proves més conegudes són **Postman** i **Insomnia**. Pel que fa a aquest projecte, les dues eines són igual d'útils i ofereixen quasi bé les mateixes possibilitats.

No obstant això, l'opció amb la qual tinc més experiència és **Postman** i, per tant, és l'escollida.

Postman [Postman 2022] (veure figura 7.12) és una plataforma d'API per a desenvolupadors que serveix per dissenyar i comprovar el funcionament de les API.



Figura 7.12: Logotip de Postman.

7.4 Base de dades

7.4.1 Entorn d'execució

En aquesta subsecció, es justificarà l'elecció de l'entorn d'execució sobre el qual s'executa l'SGBD del projecte: **Docker**.

Abans d'instal·lar i fer servir un SGBD directament sobre el sistema operatiu de l'ordinador personal, s'ha optat per fer-ho en un entorn que cada vegada és més popular i que ofereix nombrosos avantatges: **Docker**.

Docker [Docker 2022] (veure figura 7.13) és un *software* de codi obert que automatitza el desplegament d'aplicacions dins de contenidors construïts sobre el nucli de Linux. Permet disposar d'imatges predefinides d'un determinat aplicatiu per tal de poder executar-les fàcilment, com ara la imatge d'un SGBD. El punt clau d'això és que un contenidor pot allotjar un determinat programari i tota la seva configuració, de manera que tot plegat sigui portable i ràpidament executable en altres màquines i sistemes operatius.



Figura 7.13: Logotip de Docker.

7.4.2 Sistema gestor de bases de dades

En aquesta subsecció, es justificarà l'elecció del sistema gestor de bases de dades utilitzat per gestionar les dades del projecte: **MySQL**.

Per començar, si s'analitzen mínimament el tipus de dades que involucra el projecte, ràpidament es pot concloure que l'opció més òptima és decantar-se per una base de dades relacional, ja que estan molt relacionades entre sí.

Els SGBD relacionals més populars i més ben documentats són Oracle, **MySQL**, Microsoft SQL Server i PostgreSQL. No obstant això, les dues opcions candidates són Oracle i **MySQL**, ja que són les úniques amb les quals tinc experiència.

Tal com s'ha vist a la secció 7.3, l'ORM escollit per al projecte funciona només amb un conjunt limitat de SGBD, el qual no contempla Oracle. Per tant, el sistema gestor de bases de dades escollit és **MySQL**.

MySQL [Oracle 2022] (veure figura 7.14) és un sistema gestor de bases de dades relacional de codi lliure i multiusuari que funciona amb múltiples fils d'execució i utilitza el llenguatge SQL.



Figura 7.14: Logotip de MySQL.

7.4.3 Proves

En aquesta subsecció, es justificarà l'elecció de l'eina utilitzada per a la realització de proves en quant a la base de dades: **DataGrip**.

Existeixen diverses opcions que permeten verificar el correcte ús d'una base de dades gestionada, en el cas d'aquest projecte, per MySQL (veure subsecció 7.4.2): mitjançant comandes directament a una consola o utilitzant un programari específic com ara MySQL Workbench o **DataGrip**.

Tot i tenir experiència amb MySQL Workbench, m'he decantat per **DataGrip** perquè l'he trobat molt més agradable visualment, més còmode i, sobretot, més potent, ja que ofereix més funcionalitats.

DataGrip [JetBrains 2022] (veure figura 7.15) és un entorn de desenvolupament integrat que permet treballar amb bases de dades i SQL. Permet guardar consultes, modificar dades i generar diagrames, entre moltes altres funcionalitats. A més a més, és compatible amb un gran nombre de SGBD.



Figura 7.15: Logotip de DataGrip.

7.5 Eines de desenvolupament

7.5.1 Editor de codi

En aquesta subsecció, es presentarà l'editor de codi amb què s'ha desenvolupat el projecte: **Visual Studio Code**.

Visual Studio Code [Microsoft 2022] (veure figura 7.16) és un editor de codi font molt lleuger i extremadament flexible i configurable. Gràcies a les seves extensions, es pot utilitzar per desenvolupar codi en qualsevol llenguatge de programació i per obrir i interpretar una gran quantitat de tipus de fitxer.



Figura 7.16: Logotip de Visual Studio Code.

Aquest editor ha resultat extremadament útil durant el desenvolupament del projecte, ja que s'ha pogut utilitzar tant pel codi font del *front-end* i el *back-end* com pel codi de la memòria, escrit amb LaTeX (veure secció 7.6).

7.6 Confecció de la memòria

7.6.1 Elaboració del document

En aquesta subsecció, es presentarà l'eina utilitzada per a l'elaboració de la memòria del projecte: **LaTeX**.

LaTeX [LaTeX Project 2022] (veure figura 7.17) és un sistema de composició de textos orientat a la creació de documents escrits que han de presentar una alta qualitat tipogràfica. Degut a les seves característiques, és molt utilitzat en articles i llibres científics, tesis tècniques, etc.



Figura 7.17: Logotip de LaTeX.

7.6.2 Disseny de les figures

En aquesta subsecció, es presentarà l'eina utilitzada per a l'elaboració de les figures presentades a la memòria: **Diagrams.net**.

Diagrams.net [JGraph Ltd 2022] (veure figura 7.18) és una eina de codi obert que permet crear una gran varietat d'esquemes i diagrames de qualsevol mena. Per exemple, es fa servir molt freqüentment per dibuixar diagrames UML.



Figura 7.18: Logotip de Diagrams.net.

7.6.3 Diagrama de Gantt

En aquesta subsecció, es presentarà l'eina utilitzada per a l'elaboració del diagrama de Gantt presentat a la memòria (veure capítol 6): **TeamGantt**.

TeamGantt [TeamGantt 2022] (veure figura 7.19) és una plataforma web que permet confeccionar diagrames de Gantt atractius i de manera relativament còmode.



Figura 7.19: Logotip de Teamgantt.

Anàlisi i disseny del sistema

En aquest capítol, es presentarà el disseny del sistema un cop ha estat analitzat. En primer lloc, es presentaran els casos d'ús de les diferents parts de l'aplicatiu (veure secció 8.1). A més, es mostrarà el disseny de la base de dades (veure secció 8.2) i el de les interfícies d'usuari (veure secció 8.3).

8.1 Anàlisi dels casos d'ús

8.1.1 Actors implicats

En aquesta subsecció, es recolliran els actors que poden interactuar amb l'aplicació de manera externa. També s'indicaran les relacions d'especialització / generalització que estableixen.

Tal com es pot veure a la figura 8.1, els actors corresponen als diferents rols d'usuari que participen en l'aplicació. Tots aquests actors, que disposen del seu propi conjunt d'interaccions possibles amb el sistema, també hereten el d'*Usuari*, ja que en són especialitzacions.

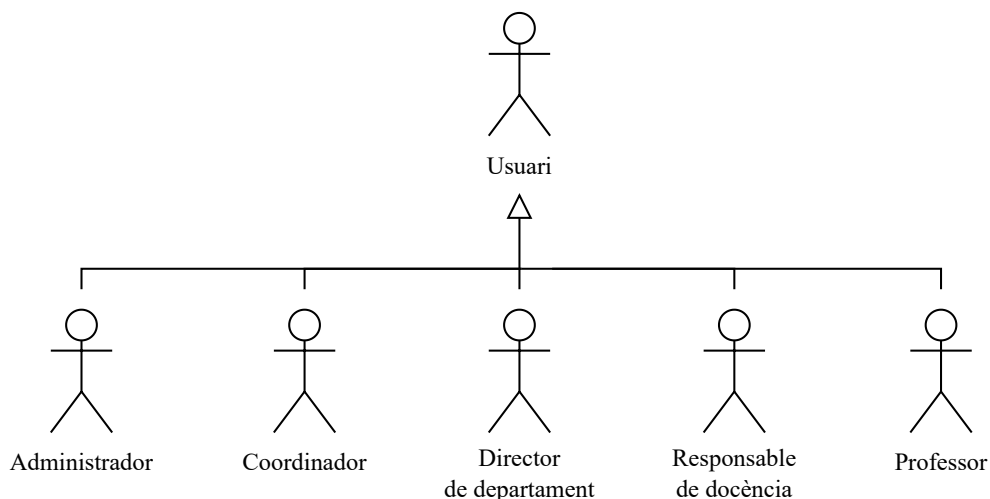


Figura 8.1: Jerarquia d'actors que poden interactuar amb l'aplicació.

8.1.2 Autenticació

En aquesta subsecció, es representaran els casos d'ús relacionats amb la part de l'autenticació d'usuaris (veure figura 8.2).

Quan es parli d'un *Token*, s'estarà fent referència a un *JSON Web Token* [IETF 2022] generat per codificar certes dades amb una finalitat concreta (més informació sobre la implementació al capítol 9).

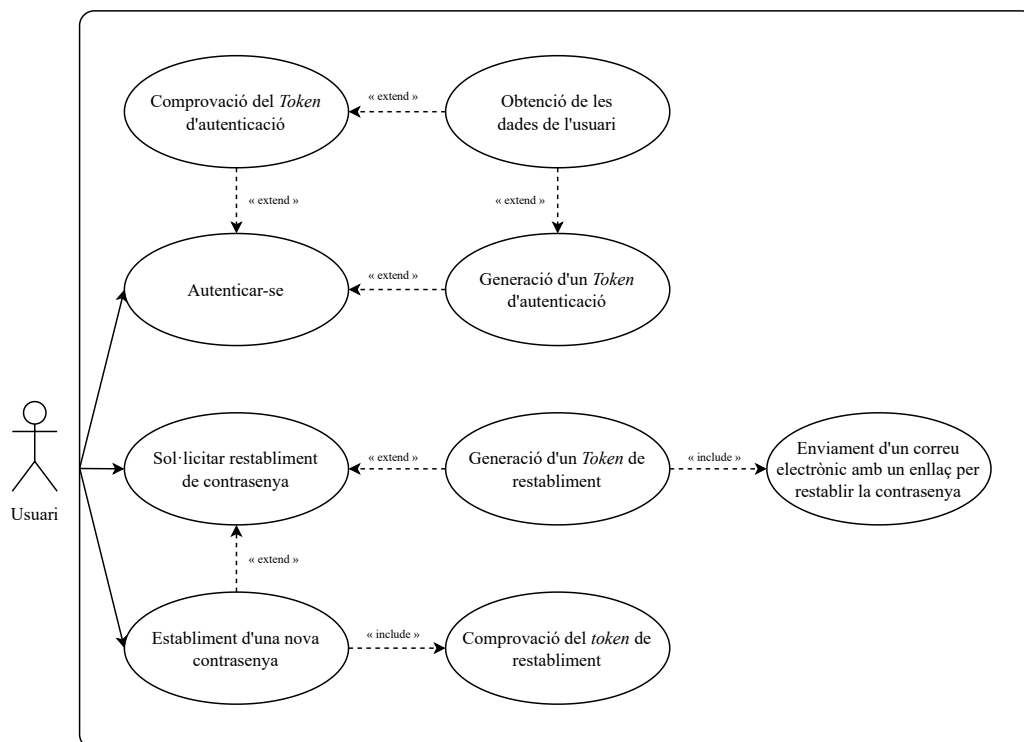


Figura 8.2: Diagrama dels casos d'ús relacionats amb l'autenticació.

8.1.3 Gestió del pla docent

8.1.3.1 General

En aquest apartat, es representaran els casos d'ús generals involucrats en la gestió del pla docent (veure figura 8.3), els quals seran desglossats en els apartats posteriors de la subsecció.

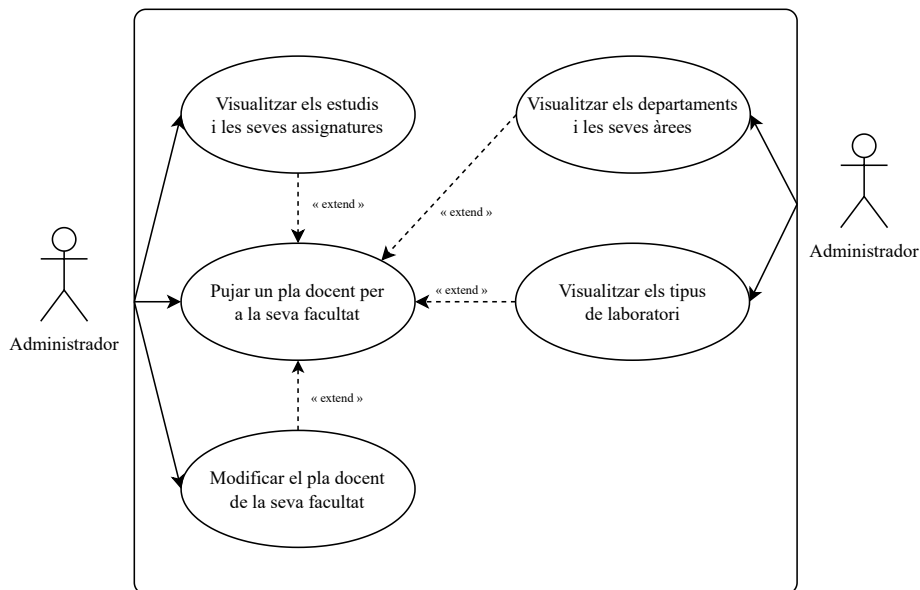


Figura 8.3: Diagrama de casos d'ús general de la gestió del pla docent.

8.1.3.2 Pujada d'un pla docent

En aquest apartat, es representaran els casos d'ús involucrats en la pujada d'un pla docent (veure figura 8.4).

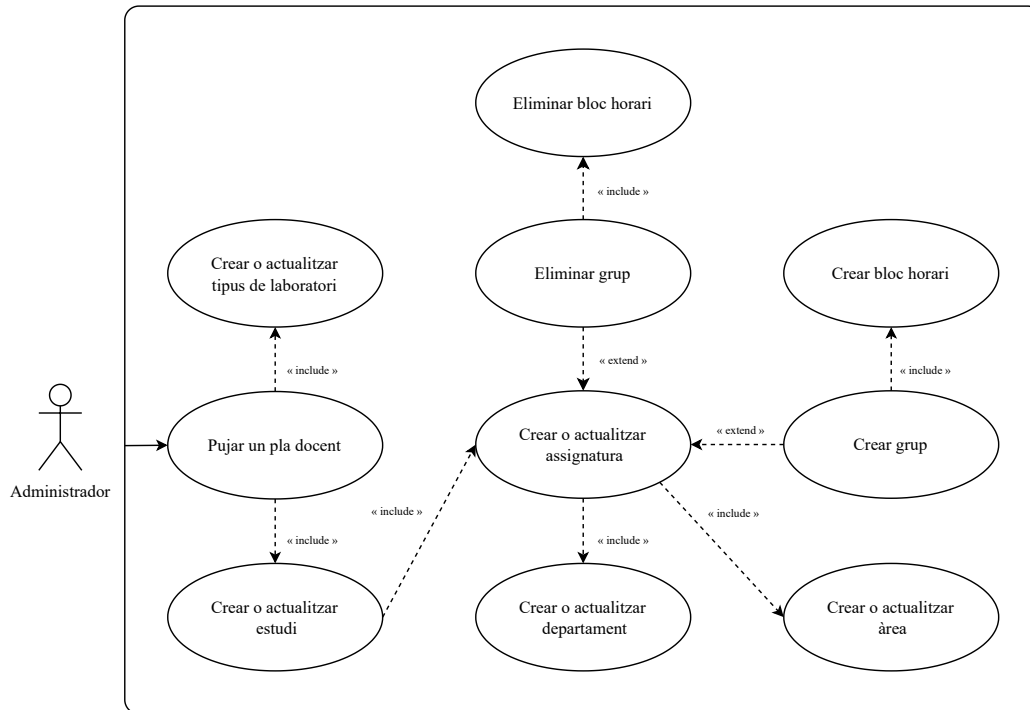


Figura 8.4: Diagrama de casos d'ús de la pujada d'un pla docent.

8.1.3.3 Modificació del pla docent

En aquest apartat, es representaran els casos d'ús involucrats en la modificació d'un pla docent (veure figura 8.5).

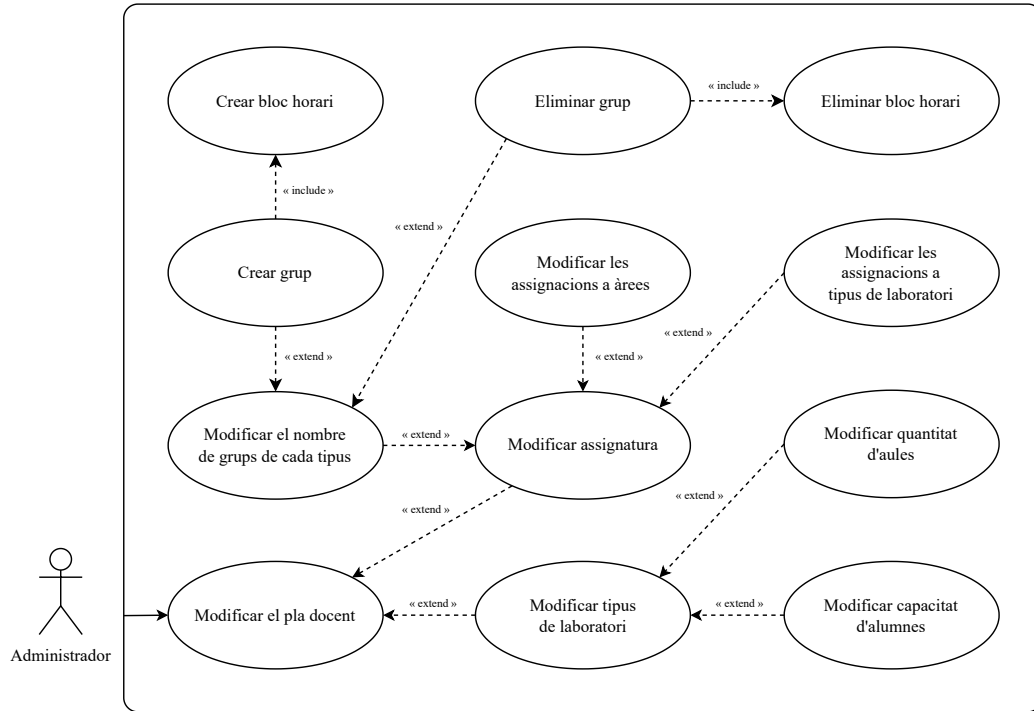


Figura 8.5: Diagrama de casos d'ús de la modificació del pla docent.

8.1.4 Gestió d'usuaris

En aquesta subsecció, es representaran els casos d'ús involucrats en la gestió d'usuaris (veure figura 8.6).

És important destacar que, per motius de llegibilitat, s'ha optat per representar només una vegada i de manera genèrica els casos d'ús "Crear usuari", "Eliminar usuari" i "Reenviar correu d'activació".

En realitat, aquests casos d'ús només s'apliquen a un usuari d'un rol concret, el qual és definit segons el rol dels usuaris que s'estiguin gestionant: en la gestió de Coordinadors, només s'apliquen a usuaris Coordinadors i així successivament.

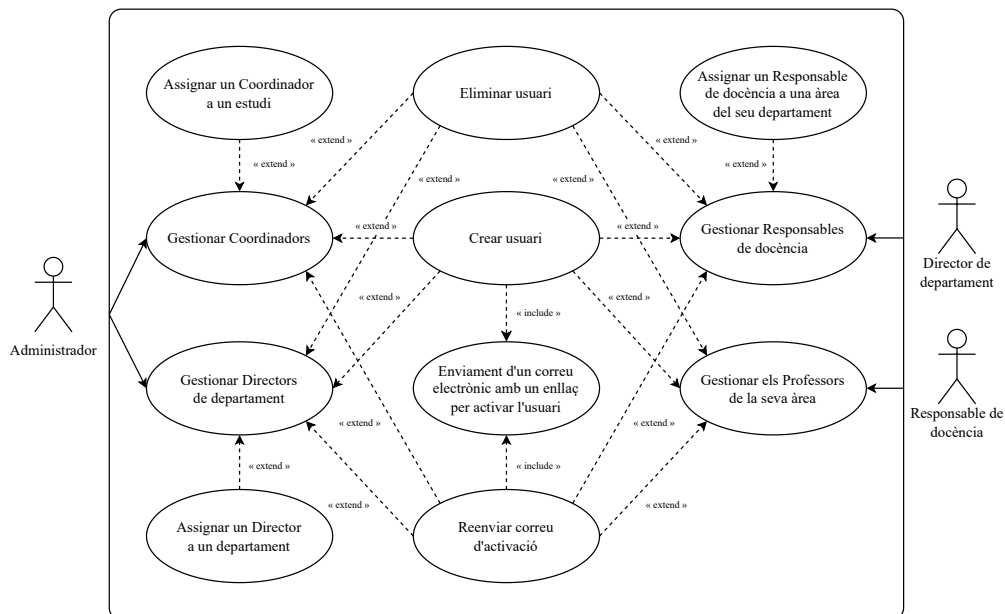


Figura 8.6: Diagrama de casos d'ús de la gestió d'usuaris.

8.1.5 Visualització d'horaris

En aquesta subsecció, es representaran els casos d'ús involucrats en la visualització d'horaris (veure figura 8.7).

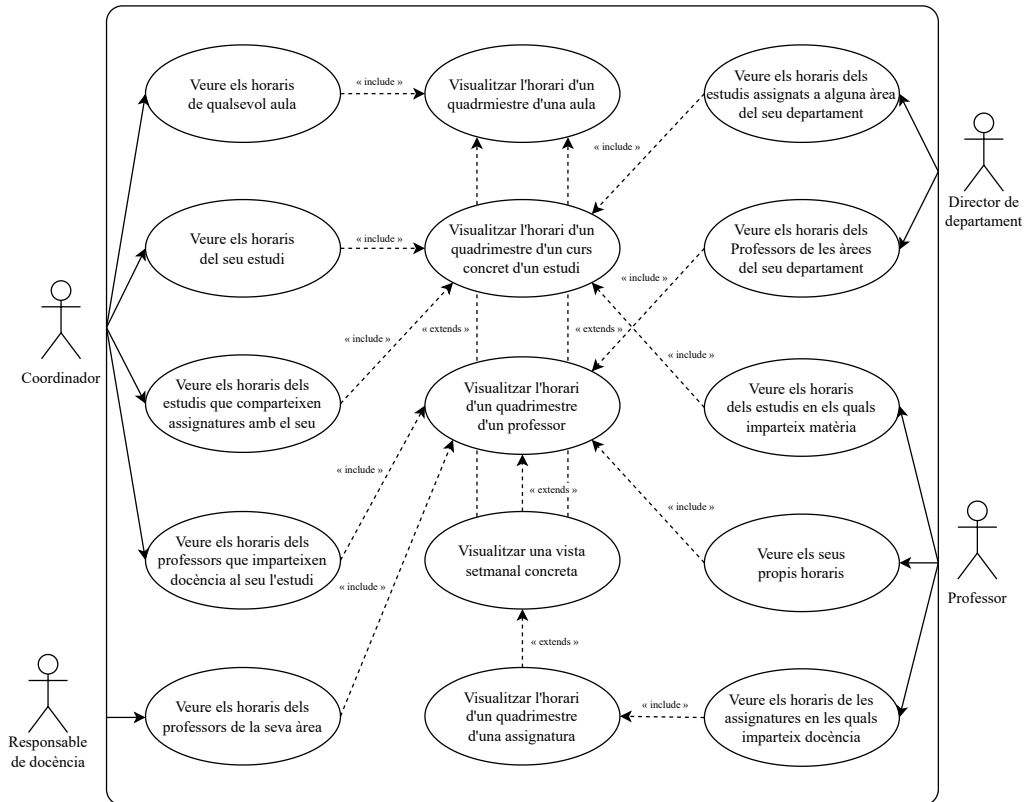


Figura 8.7: Diagrama de casos d'ús de la visualització d'horaris.

8.1.6 Modificació d'horaris d'estudis

8.1.6.1 General

En aquest apartat, es representaran els casos d'ús generals involucrats en la modificació dels horaris dels estudis (veure figura 8.8), els quals seran desglossats en els apartats posteriors de la subsecció.

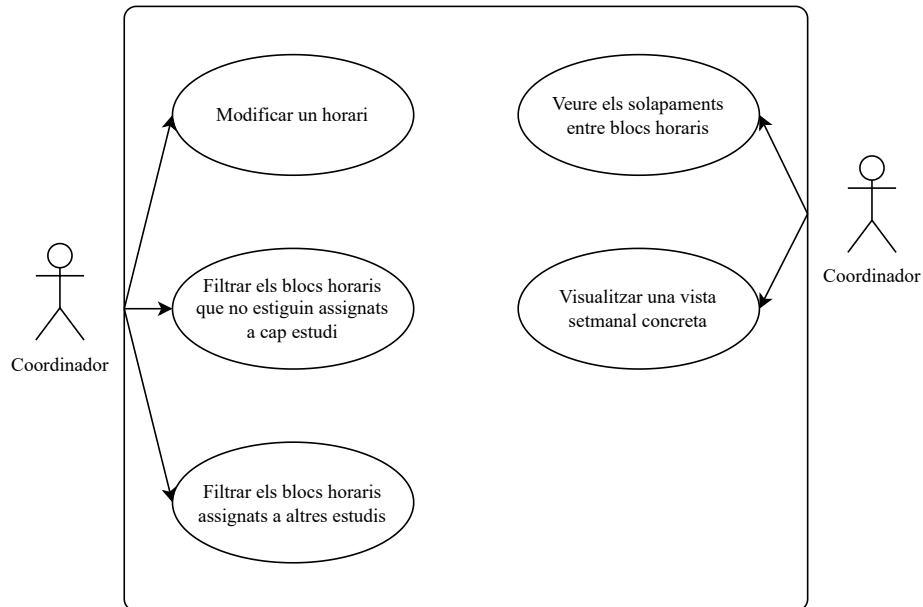


Figura 8.8: Diagrama de casos d'ús general de la modificació d'horaris.

8.1.6.2 Modificació d'un horari d'un estudi

En aquest apartat, es representaran els casos d'ús involucrats en la modificació d'un horari d'un estudi (veure figura 8.9).

És important destacar que el cas d'ús "Modificar un horari del seu estudi" es refereix a la modificació de l'horari d'un quadrimestre d'un dels cursos de l'estudi que gestiona el Coordinador en qüestió.

A més a més, cal afegir que, per motius de llegibilitat, s'ha optat per representar només una vegada i de manera genèrica el cas d'ús "Modificar dades relacionades amb el temps", ja que serveix tant per blocs horaris com per blocs horaris genèrics.

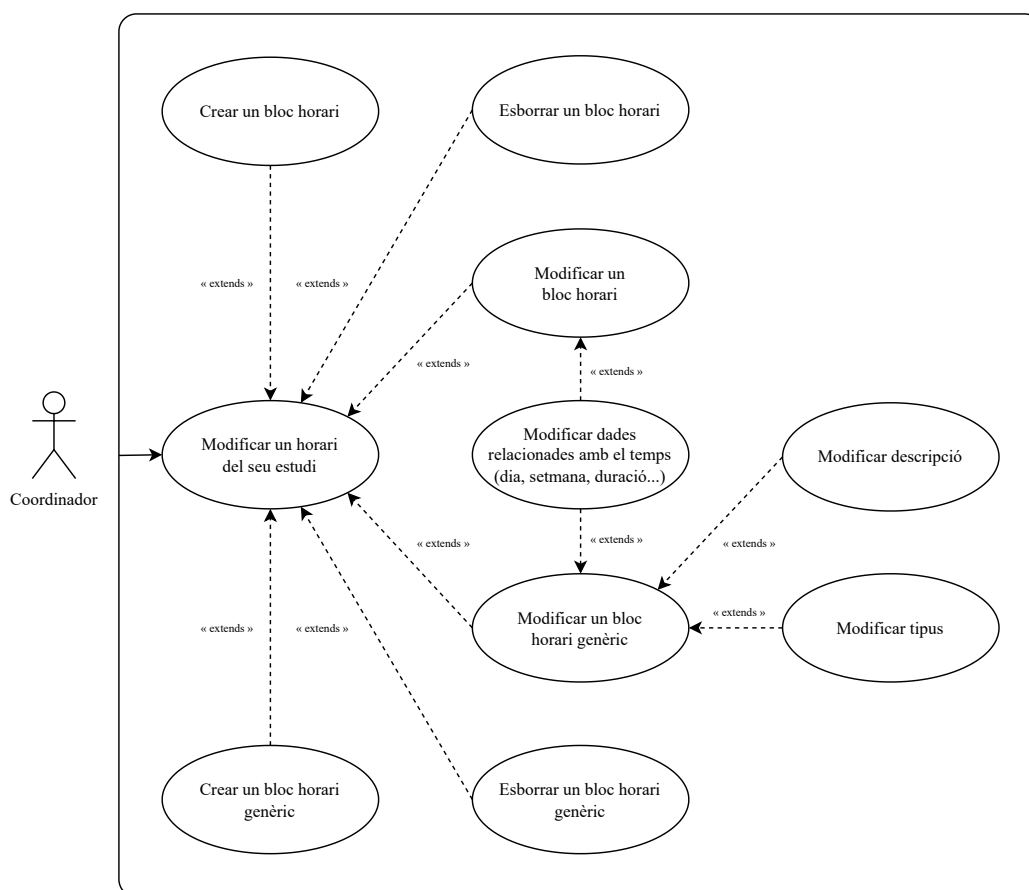


Figura 8.9: Diagrama de casos d'ús de la modificació d'un horari d'un estudi.

8.1.6.3 Visualització de solapaments

En aquest apartat, es representaran els casos d'ús involucrats en la visualització de solapaments entre blocs horaris (veure figura 8.10).

És important destacar que el cas d'ús "Veure els solapaments entre blocs horaris" es refereix als solapaments existents a l'horari d'un quadrimestre d'un dels cursos de l'estudi que gestiona el Coordinador en qüestió.

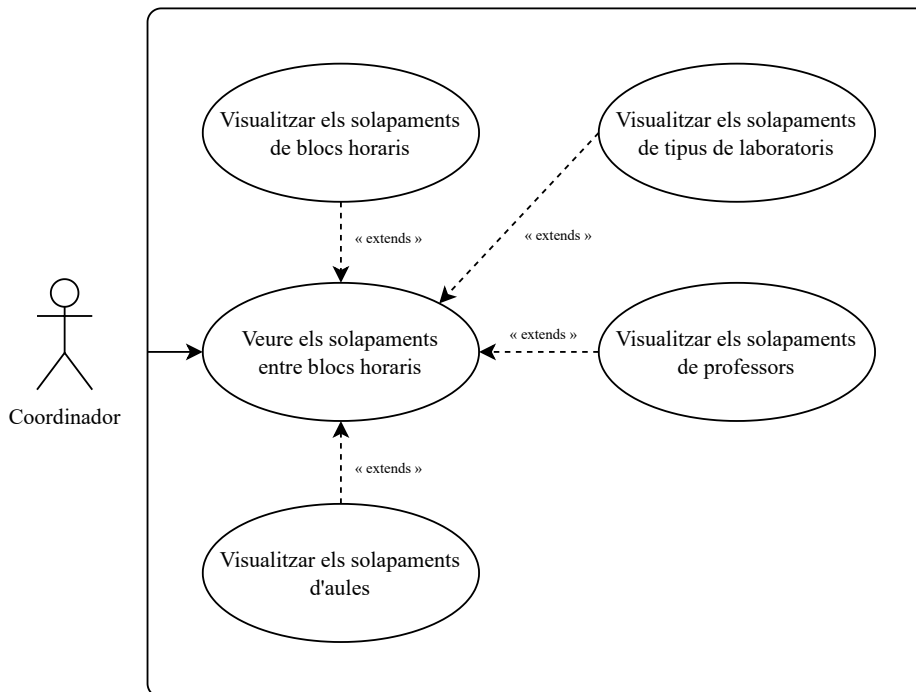


Figura 8.10: Diagrama de casos d'ús de la visualització de solapaments.

8.2 Disseny de la base de dades

8.2.1 Model entitat-relació

En aquesta subsecció, es presentarà el diagrama entitat-relació dissenyat per emmagatzemar i gestionar adequadament les dades de l'aplicació (veure figura 8.11).

Per tal d'evitar la rectificació manual del diagrama davant de qualsevol canvi en el disseny, s'ha optat per utilitzar una eina que en permeti la generació automàtica a partir de les taules que formin la base de dades. Més concretament, es tracta de DataGrip [[JetBrains 2022](#)].

La part negativa d'aquest programa és que no explicita la cardinalitat de les relacions. Per aquest motiu, és molt important remarcar que totes les relacions són “u a molts” exceptuant les següents que són “u a u”:

- Estudi (0..1) → (0..1) Usuari.
- Departament (0..1) → (0..1) Usuari.
- Àrea (0..1) → (0..1) Usuari.

També cal tenir en compte que les relacions “molts a molts” hi apareixen ja “aplicades” amb la taula intermitja entre les dues entitats en qüestió, la qual és necessària per a la seva implementació. A més a més, l'herència d'usuaris també hi apareix “resolta” amb una implementació concreta (veure capítol 7 per conèixer els detalls d'aquesta decisió).

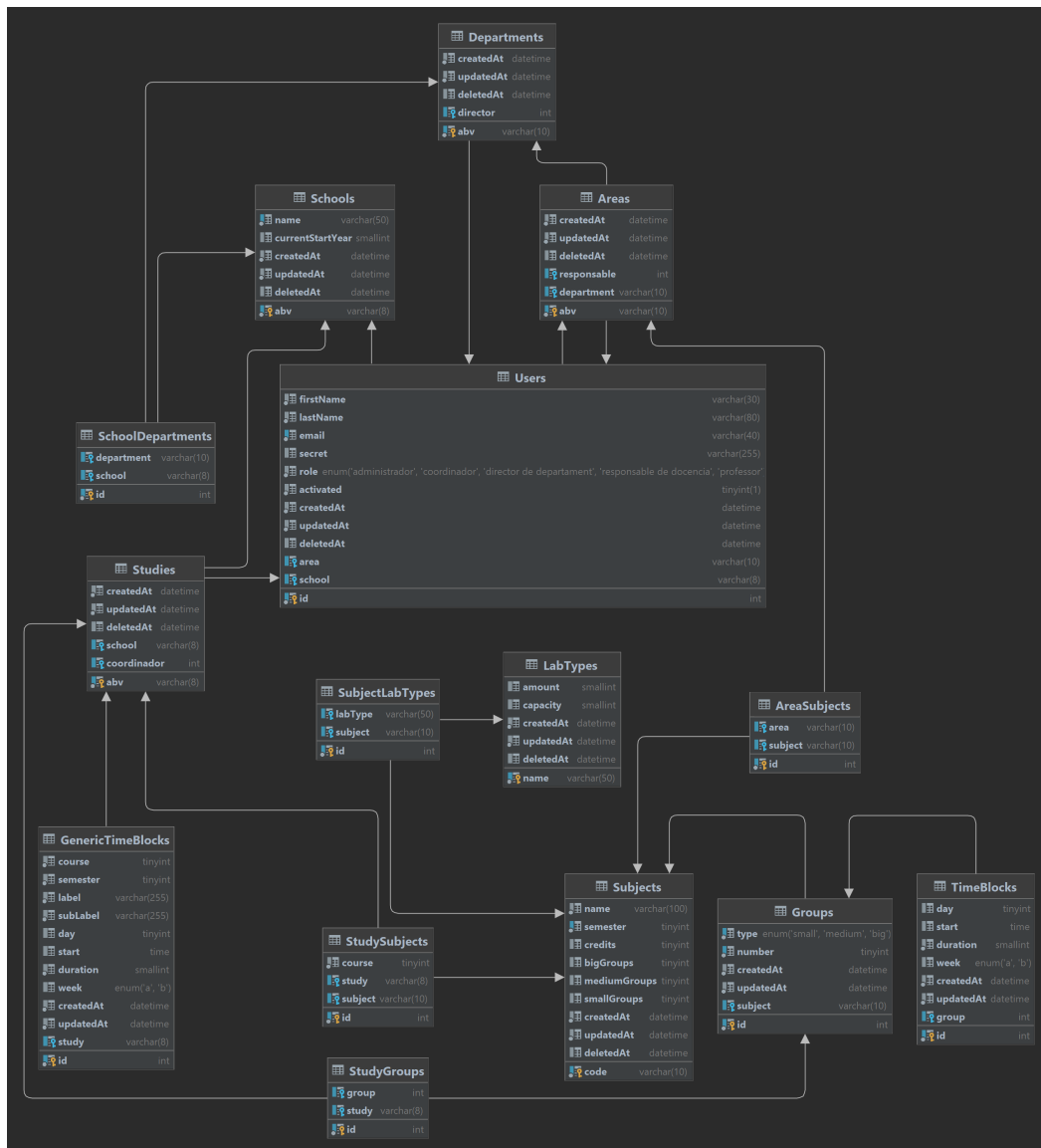


Figura 8.11: Diagrama relacional de la base de dades.

8.2.2 Descripció de les entitats

8.2.2.1 Consideracions inicials

En aquest apartat, es presentaran una sèrie de consideracions inicials amb l'objectiu de clarificar el contingut dels que segueixen.

En primer lloc, és necessari definir el format que adoptarà cadascun dels atributs de les taules:

Nom <Tipus> [Restriccions]: Descripció

D'altra banda, cal destacar que la majoria de les taules disposen de tres atributs comuns, els quals només es descriuen a continuació per tal d'evitar repeticions innecessàries:

- **createdAt** <datetime> [NOT NULL]: Moment de la creació de l'entitat.
- **updatedAt** <datetime> [NOT NULL]: Moment de l'última modificació de l'entitat.
- **deletedAt** <datetime>: Moment de l'eliminació de l'entitat.

8.2.2.2 Facultats o *Schools*

En aquest apartat, es descriuran els atributs de la taula "Facultats" ("*Schools*" al diagrama):

- **abv** <varchar(8)> [PRIMARY KEY]: Abreviació (clau primària).
- **name** <varchar(50)> [NOT NULL]: Nom.
- **currentStartYear** <smallint>: Any d'inici del curs acadèmic actual.

8.2.2.3 Usuaris o *Users*

En aquest apartat, es descriuran els atributs de la taula "Usuaris" ("*Users*" al diagrama):

- **id** <int> [PRIMARY KEY AUTOINCREMENT]: Identificador numèric (clau primària).
- **firstName** <varchar(30)> [NOT NULL]: Nom.
- **lastName** <varchar(80)> [NOT NULL]: Cognoms.
- **email** <varchar(40)> [NOT NULL]: Adreça de correu electrònic.
- **secret** <varchar(255)> [NOT NULL]: Contrasenya (encriptada).
- **role** <ENUM(<llista de rols>)> [NOT NULL]: Rol.
- **activated** <tinyint(1)> [DEFAULT 0]: Indica si l'usuari ha estat activat o no.
- **area** <varchar(10)>: En cas que el rol sigui Professor, abreviació de l'àrea a la qual pertany (clau forana).
- **school** <varchar(8)>: Abreviació de la facultat a la qual pertany (clau forana).

8.2.2.4 Departaments o *Departments*

En aquest apartat, es descriuran els atributs de la taula “Departaments” (“*Departments*” al diagrama):

- **abv** <varchar(10)> [PRIMARY KEY]: Abreviació (clau primària).
- **director** <int>: Identificador numèric de l’usuari que el dirigeix (clau forana).

8.2.2.5 FacultatDepartaments o *SchoolDepartments*

En aquest apartat, es descriuran els atributs de la taula “FacultatDepartaments” (“*SchoolDepartments*” al diagrama):

- **id** <int> [PRIMARY KEY]: Identificador numèric (clau primària).
- **school** <varchar(8)>: Abreviació de la facultat a la qual pertany (clau forana).
- **department** <varchar(10)>: Abreviació del departament al qual pertany (clau forana).

Aquesta taula també conté un índex que assegura la unicitat de la parella d’atributs “*school*” - “*department*”.

8.2.2.6 Àrees o *Areas*

En aquest apartat, es descriuran els atributs de la taula “Àrees” (“*Areas*” al diagrama):

- **abv** <varchar(10)> [PRIMARY KEY]: Abreviació (clau primària).
- **responsable** <int>: Identificador numèric de l’usuari que la gestiona (clau forana).
- **departament** <int>: Abreviació del departament al qual pertany (clau forana).

8.2.2.7 Estudis o *Studies*

En aquest apartat, es descriuran els atributs de la taula “Estudis” (“*Studies*” al diagrama):

- **abv** <varchar(8)> [PRIMARY KEY]: Abreviació (clau primària).
- **school** <varchar(8)>: Abreviació de la facultat a la qual pertany (clau forana).

- **coordinador** <int>: Identificador numèric de l'usuari que el gestiona (clau forana).

8.2.2.8 Assignatures o *Subjects*

En aquest apartat, es descriuran els atributs de la taula “Assignatures” (“*Subjects*” al diagrama):

- **code** <varchar(10)> [PRIMARY KEY]: Codi (clau primària).
- **name** <varchar(100)> [NOT NULL]: Nom.
- **semester** <tinyint> [NOT NULL]: Semestre durant el qual es cursa.
- **credits** <tinyint>: Nombre de crèdits.
- **bigGroups** <tinyint>: Nombre de grups grans que ha de tenir.
- **mediumGroups** <tinyint>: Nombre de grups mitjans que ha de tenir.
- **smallGroups** <tinyint>: Nombre de grups petits que ha de tenir.

8.2.2.9 EstudiAssignatures o *StudySubjects*

En aquest apartat, es descriuran els atributs de la taula “EstudiAssignatures” (“*StudySubjects*” al diagrama):

- **id** <int> [PRIMARY KEY]: Identificador numèric (clau primària).
- **course** <tinyint> [NOT NULL]: Curs en el qual es cursa l'assignatura a l'estudi.
- **study** <varchar(8)>: Abreviació de l'estudi al qual pertany (clau forana).
- **subject** <varchar(10)>: Codi de l'assignatura a la qual pertany (clau forana).

Aquesta taula també conté un índex que assegura la unicitat de la parella d'atributs “*study*” - “*subject*”.

8.2.2.10 ÀreaAssignatures o *AreaSubjects*

En aquest apartat, es descriuran els atributs de la taula “ÀreaAssignatures” (“*AreaSubjects*” al diagrama):

- **id** <int> [PRIMARY KEY]: Identificador numèric (clau primària).

- **area** <varchar(10)>: Abreviació de l'àrea a la qual pertany (clau forana).
- **subject** <varchar(10)>: Codi de l'assignatura a la qual pertany (clau forana).

Aquesta taula també conté un índex que assegura la unicitat de la parella d'atributs “*area*” - “*subject*”.

8.2.2.11 TipusLaboratori o *LabTypes*

En aquest apartat, es descriuran els atributs de la taula “TipusLaboratori” (“*LabTypes*” al diagrama):

- **name** <varchar(50)> [PRIMARY KEY]: Nom.
- **amount** <smallint>: Quantitat existent.
- **capacity** <smallint>: Capacitat d'alumnes.

8.2.2.12 AssignaturaTipusLaboratori o *SubjectLabTypes*

En aquest apartat, es descriuran els atributs de la taula “AssignaturaTipusLaboratori” (“*SubjectLabTypes*” al diagrama):

- **id** <int> [PRIMARY KEY]: Identificador numèric (clau primària).
- **labType** <varchar(50)>: Nom del tipus de laboratori al qual pertany (clau forana).
- **subject** <varchar(10)>: Codi de l'assignatura a la qual pertany (clau forana).

Aquesta taula també conté un índex que assegura la unicitat de la parella d'atributs “*labType*” - “*subject*”.

8.2.2.13 Grups o *Groups*

En aquest apartat, es descriuran els atributs de la taula “Grups” (“*Groups*” al diagrama):

- **id** <int> [PRIMARY KEY]: Identificador numèric (clau primària).
- **type** <enum('small', 'medium', 'big')> [NOT NULL]: Tipus.
- **number** <tinyint> [NOT NULL]: Número (numeració segons tipus).

- **subject** <varchar(10)>: Codi de l'assignatura a la qual pertany (clau forana).

8.2.2.14 EstudiGrups o *StudyGroups*

En aquest apartat, es descriuran els atributs de la taula “EstudiGrups” (“*StudyGroups*” al diagrama):

- **id** <int> [PRIMARY KEY]: Identificador numèric (clau primària).
- **group** <int>: Identificador numèric del grup al qual pertany (clau forana).
- **study** <varchar(8)>: Abreviació de l'estudi al qual pertany (clau forana).

Aquesta taula també conté un índex que assegura la unicitat de la parella d'atributs “group” - “study”.

8.2.2.15 BlocsHoraris o *TimeBlocks*

En aquest apartat, es descriuran els atributs de la taula “BlocsHoraris” (“*TimeBlocks*” al diagrama):

- **id** <int> [PRIMARY KEY]: Identificador numèric (clau primària).
- **day** <tinyint>: Dia de la setmana al qual està assignat.
- **start** <time>: Hora d'inici.
- **duration** <smallint> [NOT NULL]: Duració.
- **week** <enum('A', 'B')>: Setmana a la qual està assignat (NULL si ho està a totes).
- **group** <int>: Identificador numèric del grup al qual pertany (clau forana).

8.2.2.16 BlocsHorarisGenèrics o *GenericTimeBlocks*

En aquest apartat, es descriuran els atributs de la taula “BlocsHorarisGenèrics” (“*GenericTimeBlocks*” al diagrama):

- **id** <int> [PRIMARY KEY]: Identificador numèric (clau primària).
- **course** <tinyint> [NOT NULL]: Curs al qual pertany.
- **semester** <tinyint> [NOT NULL]: Semestre al qual pertany.

- **label** <varchar(255)> [NOT NULL]: Etiqueta.
- **subLabel** <varchar(255)> [NOT NULL]: Subetiqueta.
- **day** <tinyint>: Dia de la setmana al qual està assignat.
- **start** <time>: Hora d'inici.
- **duration** <smallint> [NOT NULL]: Duració.
- **week** <enum('A', 'B')>: Setmana a la qual està assignat (NULL si ho està a totes).
- **study** <varchar(8)>: Abreviació de l'estudi al qual està assignat (clau forana).

8.3 Interfícies d'usuari

En aquesta secció, s'exposarà el disseny de les diferents finestres que conformaran l'aplicació web client.

Cal destacar que l'aparença final de l'aplicació no ha de ser necessàriament igual que la que es mostrarà a les subseccions següents.

8.3.1 Interfícies d'autenticació

En aquesta subsecció, es presentarà el disseny de les finestres més importants relacionades amb la part de l'autenticació.

A la figura 8.12 es pot veure el disseny de la finestra en la qual un usuari pot autenticar-se o bé indicar que ha oblidat la seva contrasenya.

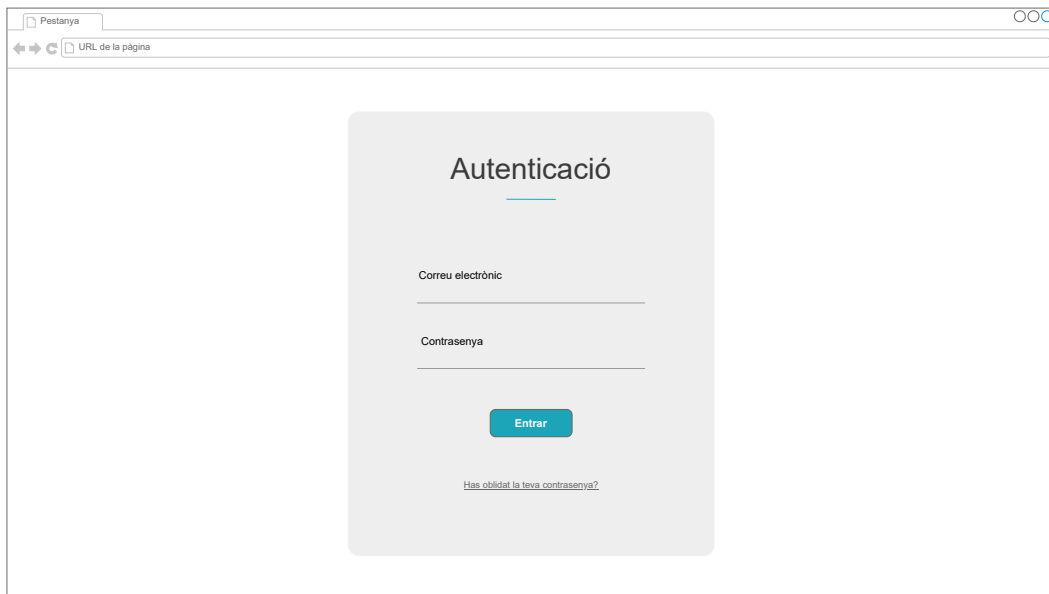


Figura 8.12: Disseny de la interfície d'autenticació.

A la figura 8.13 es pot veure el disseny de la finestra en la qual un usuari pot introduir la seva adreça de correu electrònic per tal de restablir la seva contrasenya o bé tornar a la finestra d'autenticació.

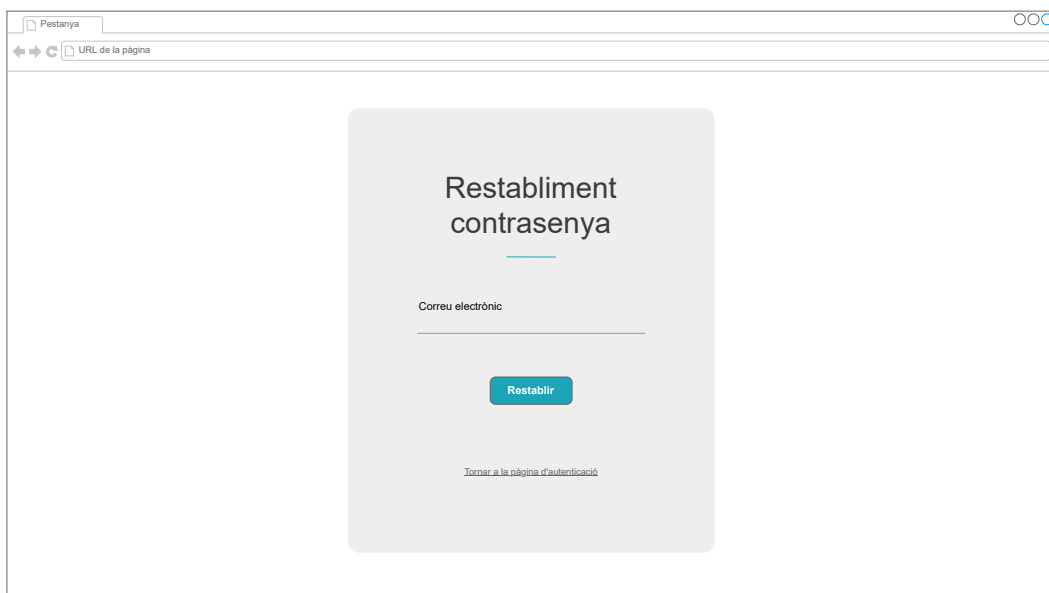


Figura 8.13: Disseny de la interfície de restabliment de contrasenya.

A la figura 8.14 es pot veure el disseny de la finestra en la qual s'informa l'usuari que se li ha enviat un correu electrònic per restablir la seva contrasenya.

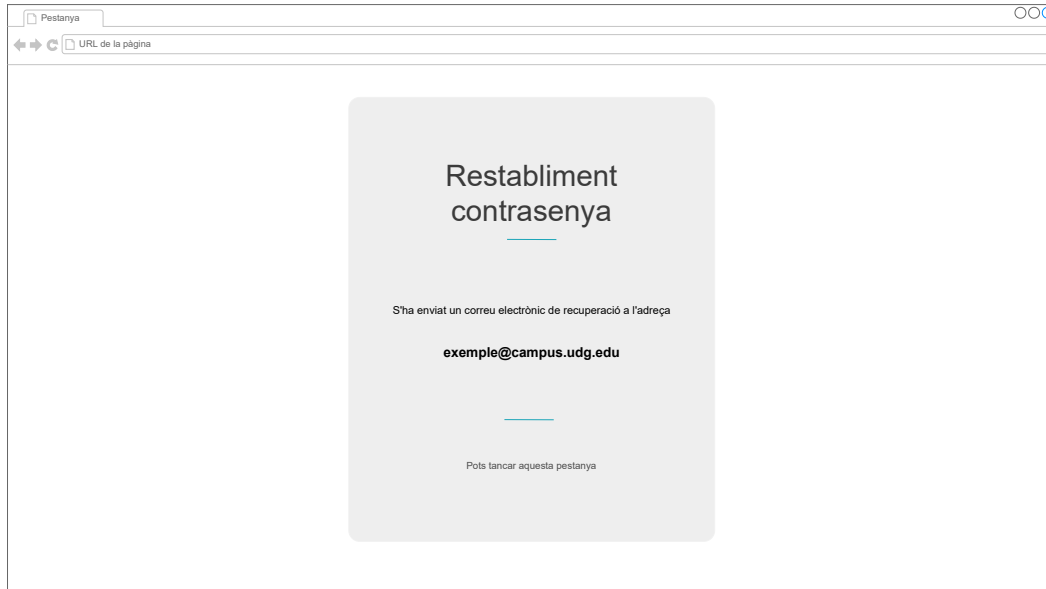


Figura 8.14: Disseny de la interfície de confirmació d'enviament d'un *email* per al restabliment de la contrasenya.

A la figura 8.15 es pot veure el disseny de la finestra en la qual l'usuari pot escollir una nova contrasenya.

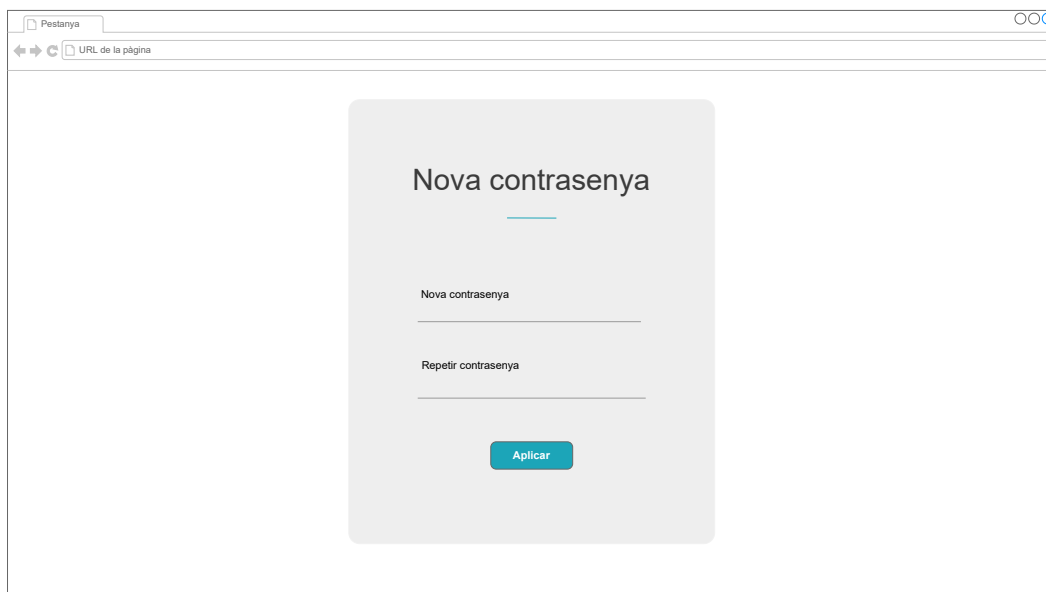


Figura 8.15: Disseny de la interfície d'elecció d'una nova contrasenya.

8.3.2 Interfícies de gestió del pla docent

En aquesta subsecció, es presentarà el disseny de les finestres més importants involucrades en la gestió del pla docent.

A la figura 8.16 es pot veure el disseny de la finestra en la qual un Administrador pot pujar el fitxer d'un pla docent per a la seva facultat i iniciar així el següent curs acadèmic.

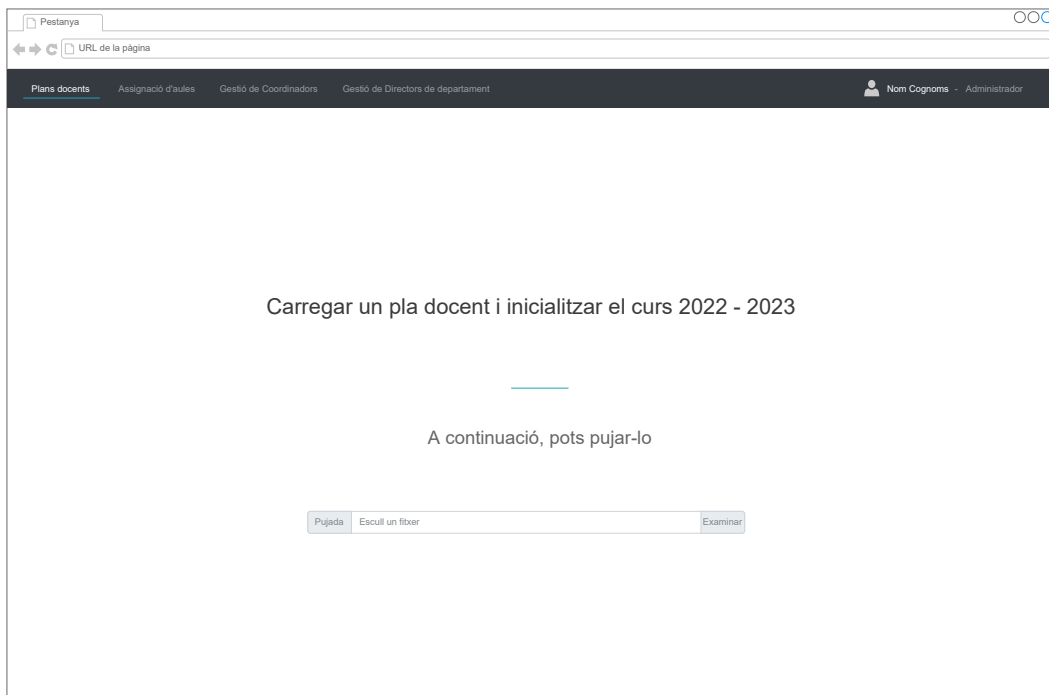


Figura 8.16: Disseny de la interfície de la pujada d'un pla docent.

A la figura 8.17 es pot veure el disseny de la finestra en la qual un Administrador pot consultar la informació general del pla docent actual, accedir a la modificació de les dades pertinents o bé anar a la finestra de pujada d'un pla docent per tal d'iniciar el curs següent.

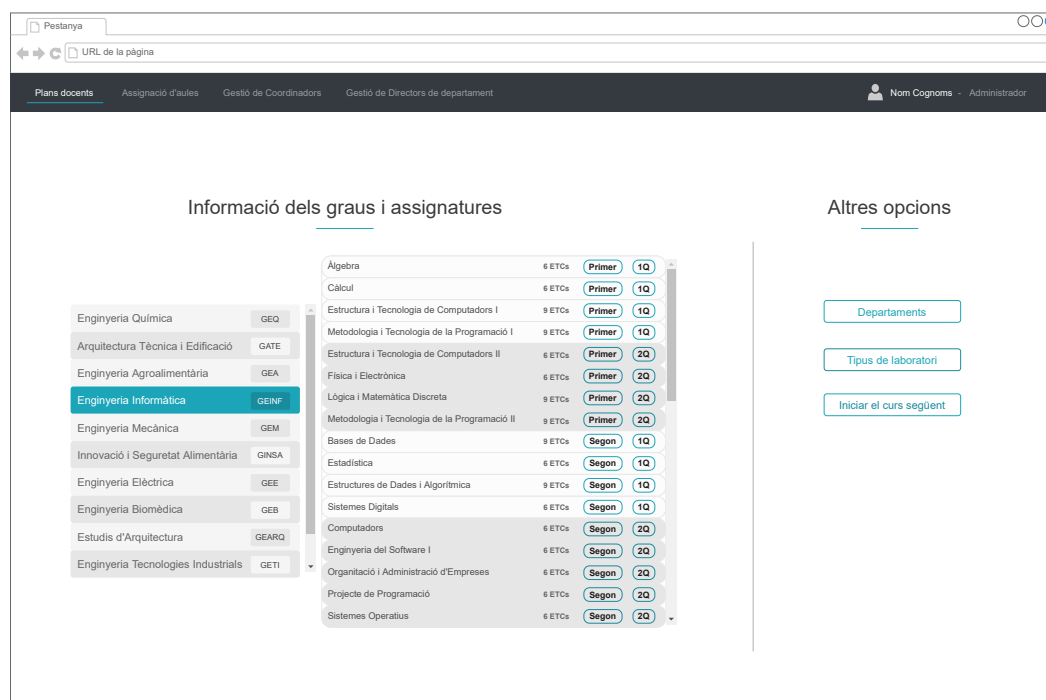


Figura 8.17: Disseny de la interfície de gestió del pla docent.

8.3.3 Interfícies de gestió d'usuaris

En aquesta subsecció, es presentarà el disseny de les finestres més importants involucrades en la gestió d'usuaris.

Aquests dissenys corresponen a la gestió de Coordinadors i la seva assignació a estudis. No obstant això, també són vàlids per a la gestió de Directores i la seva assignació a departaments, per a la gestió de Responsables de docència i la seva assignació a àrees i per a la gestió de Professors.

A la figura 8.18 es pot veure el disseny de la finestra en la qual un usuari Administrador pot gestionar l'assignació d'usuaris Coordinadors a estudis o bé accedir al seu control.

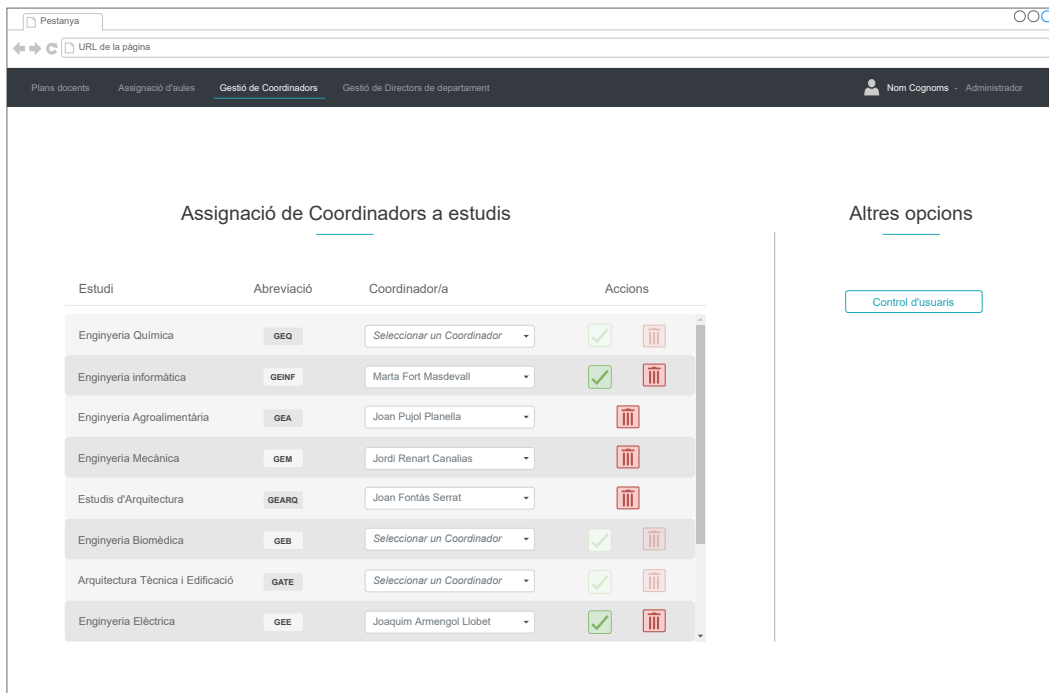


Figura 8.18: Disseny de la interfície d'assignació de Coordinadors a estudis.

A la figura 8.18 es pot veure el disseny de la finestra en la qual un usuari Administrador pot controlar usuaris Coordinadors. Més concretament, pot crear-ne, eliminar-ne o bé reenviar el correu electrònic d'activació als que no estiguin activats.

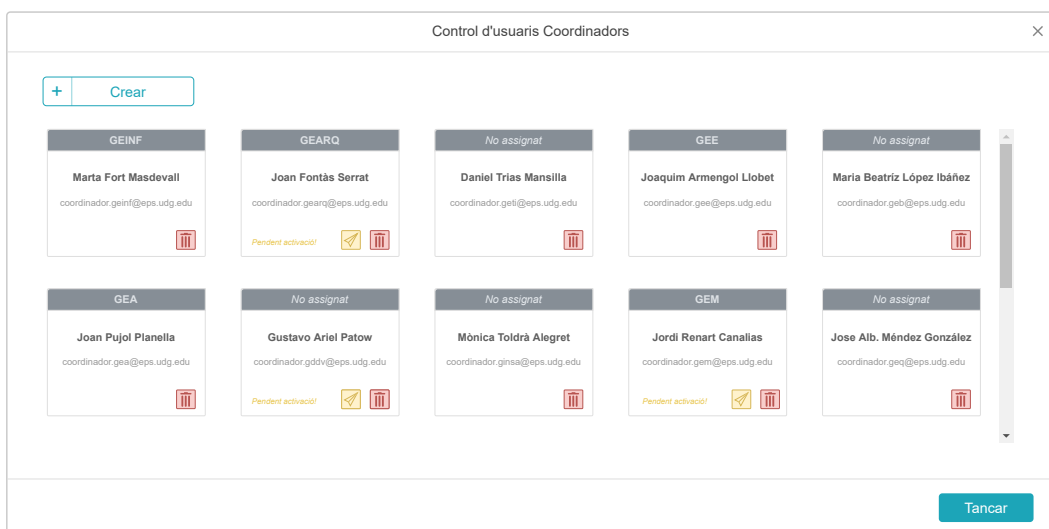
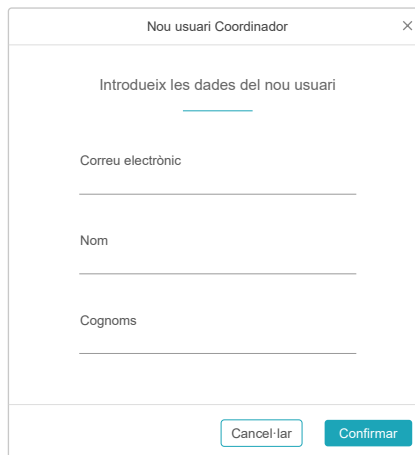


Figura 8.19: Disseny de la interfície de control de Coordinadors.

A la figura 8.20 es pot veure el disseny de la finestra en la qual un usuari Administrador pot introduir les dades necessàries per crear nous usuaris Coordinadors.



Nou usuari Coordinador

Introdueix les dades del nou usuari

Correu electrònic

Nom

Cognoms

Cancel·lar Confirmar

The image shows a modal dialog box titled "Nou usuari Coordinador" with a close button (X) in the top right corner. The main content area contains the instruction "Introdueix les dades del nou usuari" followed by three input fields labeled "Correu electrònic", "Nom", and "Cognoms". At the bottom of the dialog, there are two buttons: "Cancel·lar" and "Confirmar".

Figura 8.20: Disseny de la interfície de creació de Coordinadors.

8.3.4 Interfícies de consulta i elaboració d'horaris

En aquesta subsecció, es presentarà el disseny de les finestres més importants involucrades en la consulta i l'elaboració d'horaris.

A la figura 8.21 es pot veure el disseny de la finestra en la qual un usuari Coordinador pot accedir a les finestres de visualització o de modificació d'un horari del grau que gestiona o bé a la selecció d'un horari dels estudis els quals comparteixen alguna assignatura amb el seu per tal de visualitzar-lo.

Aquest disseny també és vàlid per a l'accés a la visualització dels horaris dels estudis corresponents per part d'un Director de departament o d'un Professor.

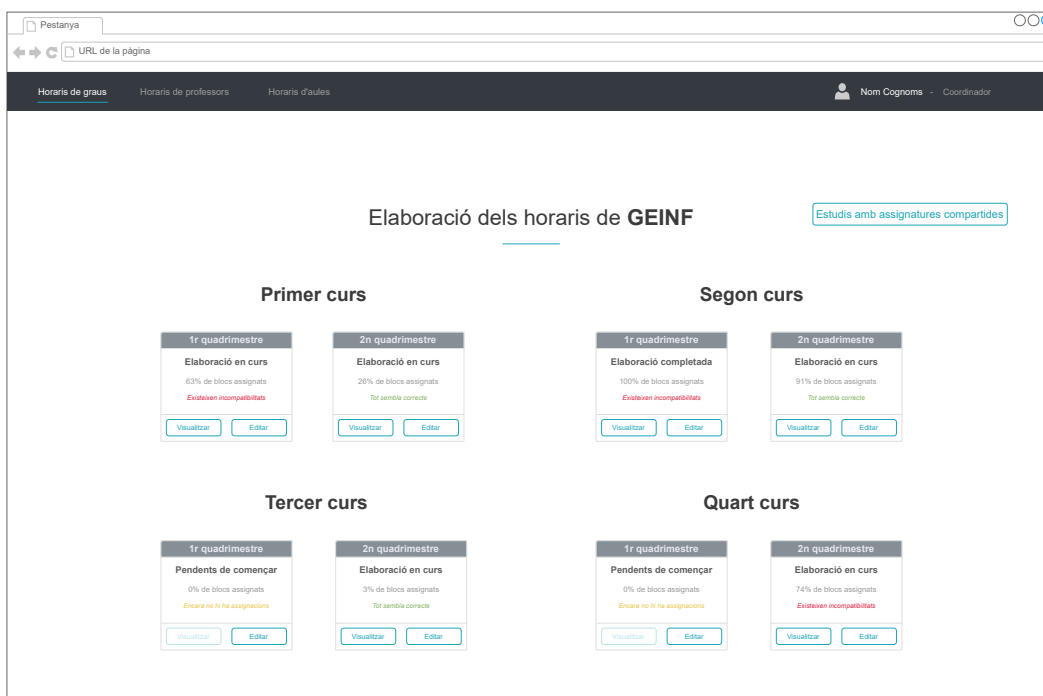


Figura 8.21: Disseny de la interfície d'accés a la gestió d'un horari d'un estudi.

A la figura 8.22 es pot veure el disseny de la finestra en la qual un usuari Coordinador pot accedir a la visualització d'un horari d'un dels Professors que imparteixen docència al seu grau.

Aquest disseny també és vàlid per a l'accés a la visualització dels horaris dels professors corresponents per part d'un Director de departament o d'un Responsable de docència.

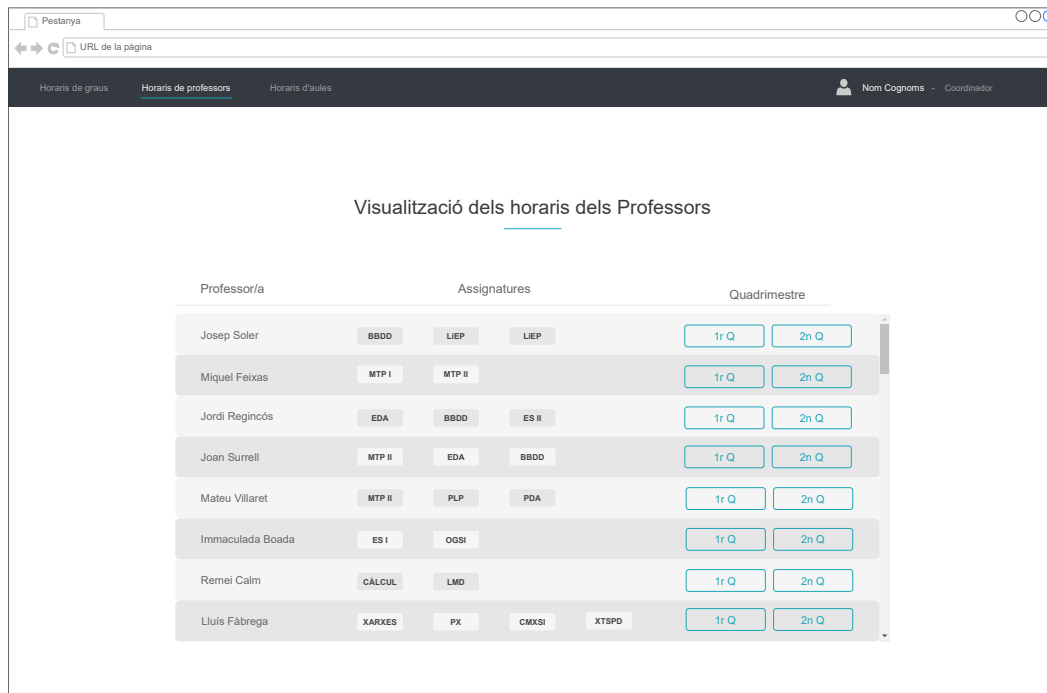


Figura 8.22: Disseny de la interfície d'accés a la visualització d'un horari d'un Professor.

A la figura 8.23 es pot veure el disseny de la finestra en la qual un usuari Coordinador pot accedir a la visualització d'un horari d'una aula.

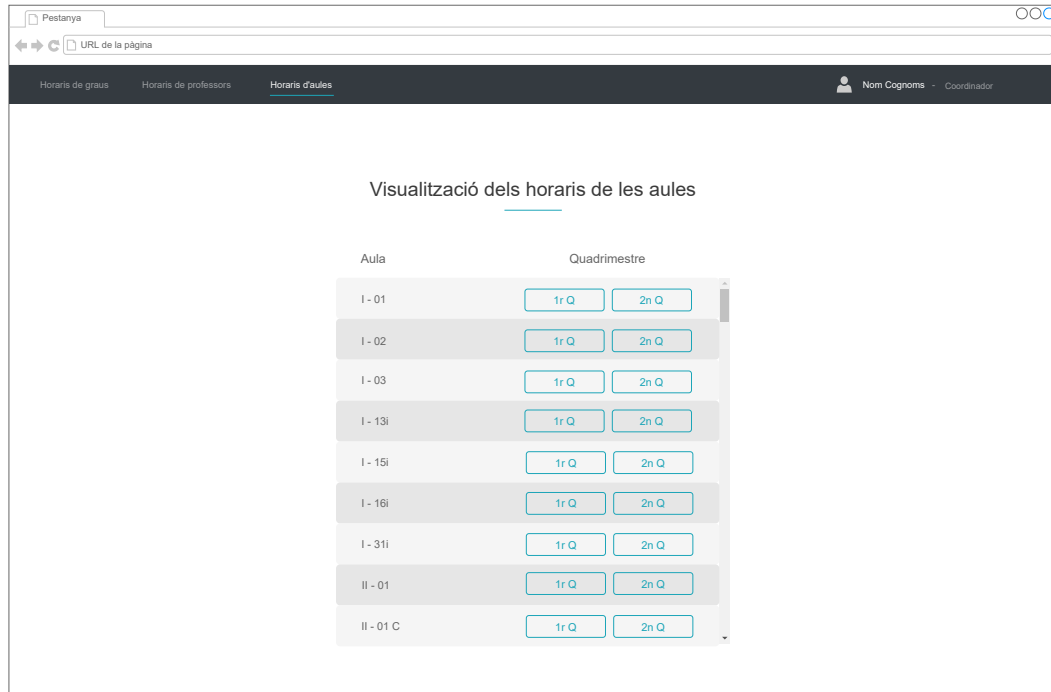


Figura 8.23: Disseny de la interfície d'accés a la visualització d'un horari d'una aula.

A la figura 8.24 es pot veure el disseny de la finestra en la qual un usuari Coordinador visualitza un horari d'un estudi. A més a més, pot alternar entre les diferents vistes setmanals, accedir al detall d'un bloc horari i tornar a la finestra de selecció d'horaris.

Aquest disseny també és vàlid per a la visualització d'horaris tant de Professors com d'aules.

Cal destacar que s'ha tingut molt en compte el fet que es pugui visualitzar tota la informació necessària sense haver de fer cap clic, cosa que ha resultat notablement difícil a causa de la gran quantitat de dades a distribuir en un espai bastant limitat.

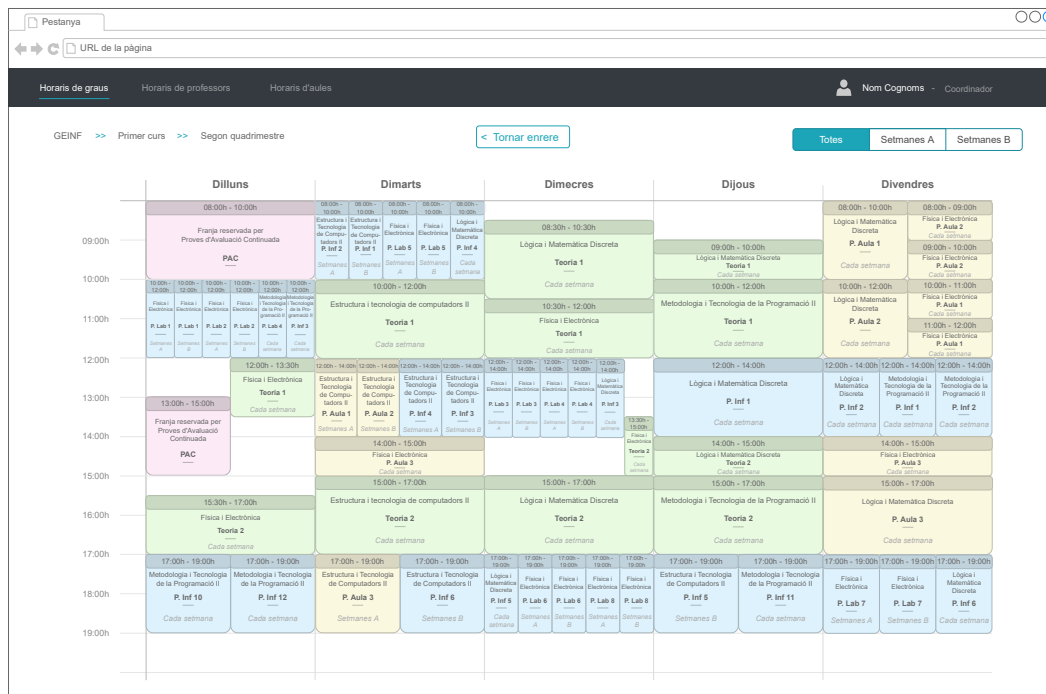


Figura 8.24: Disseny de la interfície de visualització d'un horari d'un estudi.

A la figura 8.25 es pot veure el disseny de la finestra en la qual un usuari Coordinador visualitza un horari d'un estudi i pot realitzar-hi modificacions: pot crear blocs horaris, arrossegar-ne, canviar-ne la mida, la setmana, etc. A més a més, pot accedir al detall d'un bloc horari i modificar-lo des d'allà. Finalment, pot accedir a un menú d'opcions per modificar certes configuracions, com ara habilitar o deshabilitar opcions de filtratge, de visualització de solapaments, etc.

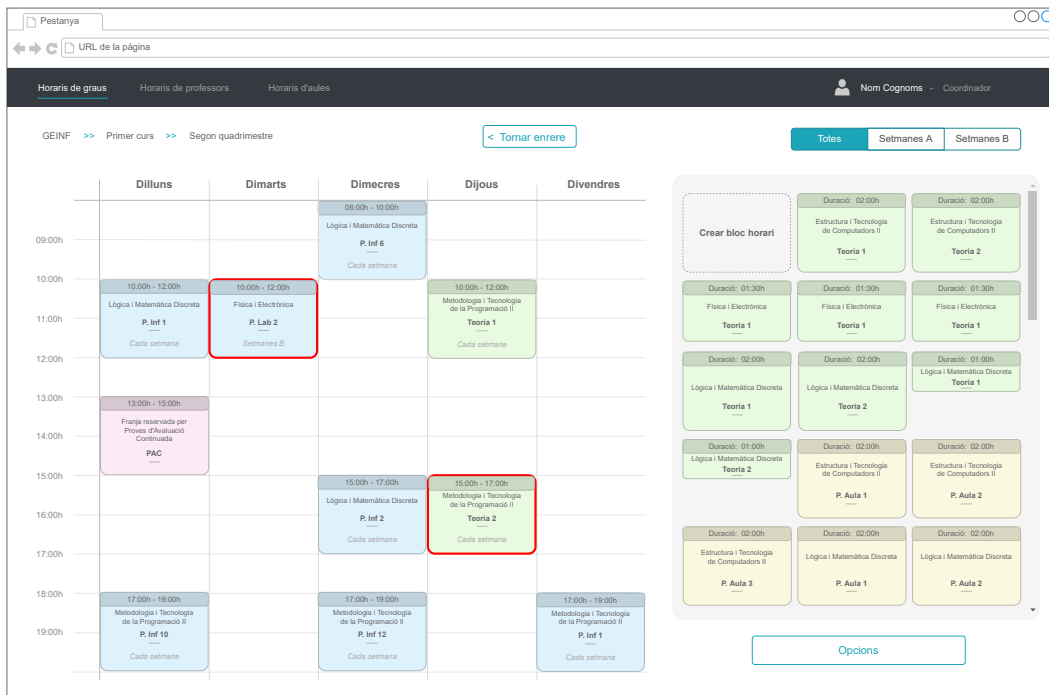


Figura 8.25: Disseny de la interfície de modificació d'un horari d'un estudi.

Implementació i proves

En aquest capítol, es presentaran alguns dels aspectes més rellevants en quant a la implementació i a les proves del programari. Més concretament, es tractaran els de l'aplicació servidor o *back-end* (veure secció 9.1) i els de l'aplicació client o *front-end* (veure secció 9.2).

9.1 Aplicació servidor

9.1.1 Configuració, variables d'entorn i *logging*

En aquesta subsecció, s'explicarà com s'ha implementat la configuració de l'aplicació servidor, com es defineixen les variables d'entorn i com és el sistema de *logs*.

Tota aplicació d'aquest estil necessita guardar informació de configuració, com ara dades de la connexió amb la base de dades, adreces de correu electrònic o qualsevol altra informació més específica.

Escriure aquestes dades de configuració directament al codi és una mala pràctica. A més a més, poden variar segons l'entorn (o *environment*) en el que s'estigui executant l'aplicació, ja sigui desenvolupament, producció, etc. Degut a això, s'ha utilitzat un paquet anomenat `config` que permet definir la configuració per a cada entorn en fitxers diferents, de manera que des del codi només s'ha d'indicar l'identificador d'un paràmetre per tal d'obtenir-ne el valor corresponent (en funció de l'entorn).

No obstant això, hi ha paràmetres molt sensibles que, el fet deixar-los en aquests fitxers de configuració, suposaria una gran vulnerabilitat de seguretat. Les dades més crítiques, com ara claus d'enciptació o credencials de la base de dades, es guarden a les variables d'entorn de la màquina en què s'executi l'aplicació.

Per aquest motiu, el paquet `config` també permet mapejar paràmetres amb el noms de les variables d'entorn corresponents. Mitjançant això, les claus d'enciptació dels *tokens* i les credencials de la base de dades de l'aplicació es guarden en variables d'entorn.

A la figura 9.1 es pot veure un fragment del fitxer (JSON) de configuració que s'aplica a l'entorn de desenvolupament.

```
"clientHost": ["http://localhost:3000"],
"email": {
  "server": {
    "host": "smtp.gmail.com",
    "port": 465
  },
  "auth": {
    "user": "test.pfghoraris@gmail.com"
  },
  "templates": {
    "resetPassword": "resetPassword",
    "emailConfirmation": "emailConfirmation"
  }
},
"jwt": {
  "auth": {
    "fields": ["id", "role"],
    "expiration": "14d"
  },
  "resetPassword": {
    "fields": ["id"],
    "expiration": "0.5 hrs"
  },
  "emailConfirmation": {
    "fields": ["id"],
    "expiration": "30d"
  }
},
}
```

Figura 9.1: Fragment del fitxer de configuració aplicat a l'entorn de desenvolupament.

Per raons de traçabilitat i depuració, s'han implementat diversos sistemes de *logging* que permet registrar informació i escriure-la tant en fitxers com a la consola.

En primer lloc, s'ha utilitzat un paquet anomenat *morgan* que registra a la consola totes les peticions rebudes i la seva resposta.

També s'ha utilitzat *debug*, un paquet que ha permès mostrar els missatges desitjats a la consola en un format agradable. A més a més, el paquet es pot activar o desactivar a través de variables d'entorn.

Finalment, s'ha fet servir el paquet *winston* per crear fitxers de *log* dels aspectes

que s'han considerat més importants, com ara els errors i les comandes SQL enviades a la base de dades.

9.1.2 Seguretat i autenticació

En aquesta subsecció, es destacaran els aspectes més importants relacionats amb la implementació dels mecanismes de seguretat i els processos d'autenticació.

A continuació s'explicarà, pas a pas, com és exactament el procés d'autenticació i se'n donaran detalls de la implementació.

1. El client fa una petició d'autenticació al servidor. Concretament, ho fa enviant-li les credencials de l'usuari mitjançant una crida POST a la ruta `/api/auth`.
2. El servidor rep la petició i la comença a processar. Primer comprova que el format de les credencials sigui el correcte. A la figura 9.2 es pot veure com es defineix el format d'aquestes dades mitjançant el paquet `joi`. Concretament, s'hi està estipulant que tant el correu com la contrasenya no poden estar buits i que el format del correu ha de ser vàlid.

```
const authValidationSchema = Joi.object({
  email: Joi.string().email().required(),
  secret: Joi.string().required()
});
```

Figura 9.2: Definició del format de les credencials mitjançant `Joi`.

3. Tot seguit, s'obtenen les dades de l'usuari de la base de dades i es comprova que realment existeixi i que estigui activat.
4. Després, es compara la contrasenya real de l'usuari amb la que s'havia rebut del client. És important destacar que la contrasenya real es guarda encriptada mitjançant el paquet `bcrypt`. A la figura 9.3 es pot veure com es compara la contrasenya rebuda (`authData.secret`) amb la contrasenya real encriptada (`user.secret`). A més, a la figura 9.4 es pot veure com es fa l'encriptació de la contrasenya (`value`). El segon paràmetre fa referència a la complexitat de l'encriptació (12 és més que suficient).

```
const validSecret = await bcrypt.compare(authData.secret, user.secret);
```

Figura 9.3: Comparació d'una contrasenya amb una contrasenya encriptada mitjançant `bcrypt`.

```
bcrypt.hashSync(value, 12)
```

Figura 9.4: Encriptació d'una contrasenya mitjançant bcrypt.

5. En aquest punt, ja es sap que totes les dades són correctes i es procedeix a generar un *token* d'autenticació. Més concretament, es tracta d'un JWT (*Json Web Token*) [IETF 2022] que, mitjançant una clau definida, en aquest cas, a les variables d'entorn, permet encriptar un conjunt de dades juntament amb una temps d'expiració. La cadena de caràcters resultant s'anomena *token* i es fa servir molt per mantenir sessions d'usuari sense estat, ja que s'emmagatzema només al client. A la figura 9.5 es pot veure com es genera un *token* mitjançant el paquet `jwt`. Cal aclarir que la funció "sign" rep tres paràmetres (dades, clau i opcions) que s'obtenen de la configuració, a partir del tipus (`type`) de *token* que es vulgui generar (autenticació, restabliment de contrasenya o activació de correu). En el cas de l'autenticació, només s'hi guarda l'identificador de l'usuari.

```
const generateJsonWebToken = (user, type) =>
  jwt.sign(
    _.pick(user, config.get(`jwt.${type}.fields`)),
    config.get(`jwt.${type}.key`),
    { expiresIn: config.get(`jwt.${type}.expiration`) }
  );
```

Figura 9.5: Generació d'un JWT mitjançant `jwt`.

6. Per acabar, tant les dades de l'usuari com el *token* generat s'envien com a resposta al client, que guarda el *token* al *local storage* del navegador per tal de mantenir la sessió iniciada.

A partir de que l'usuari està autenticat, totes les peticions que el client faci a l'API contindràn una camp anomenat "Authorization" a la capçalera. El valor d'aquest camp serà el *token* d'autenticació que s'havia obtingut en el procés d'autenticació. Més concretament, la convenció diu que davant del *token* hi ha de constar la paraula "Bearer".

Davant de qualsevol petició a l'API, el servidor comprova la presència i la validesa aquest *token*. Gràcies a això, es disposa d'un mecanisme d'autorització d'ús de l'API que evita peticions no desitjades. A continuació, es detallarà com és exactament aquest procés.

1. El servidor rep una petició que requereix autenticació (pràcticament totes) i n'extreu el *token* que conté el camp "Authorization" de la capçalera.

2. Seguidament, procedeix a validar-lo i a desencriptar-ne les dades (identificador de l'usuari). A la figura 9.6 es pot veure com es realitza aquesta validació a través del paquet `jwt`. La clau d'encryptació que fa servir és la mateixa que en la generació, guardada a les variables d'entorn.

```
User.verifyAuthToken = bearerToken => {
  const [type, token] = bearerToken.split(' ');
  if (type !== 'Bearer' || !token) {
    throw new Error('Invalid token type');
  }
  return jwt.verify(token, config.get('jwt.auth.key'));
};
```

Figura 9.6: Validació d'un *token* d'autenticació mitjançant `jwt`.

3. A continuació, s'intenta recuperar l'usuari en qüestió de la base de dades. Si l'usuari existeix i està activat, es crida la següent funció de la línia de processament de peticions. Gràcies a això, si un usuari que està autenticat és eliminat, no podrà realitzar cap més operació, ja que a la que faci una crida més al servidor, se li tancarà la sessió.

En darrer lloc, cal remarcar que, com a mesura de seguretat, s'ha implementat un mecanisme de validació d'accés a les diferents rutes del client. Tal com s'ha vist al capítol 5, cada rol d'usuari disposa d'un conjunt de funcionalitats principals. Com que cadascuna correspon a una ruta diferent (`/plans-docents`, `/horaris-graus`, etc.), el problema pot sorgir quan un usuari intenta accedir a una ruta que formi part d'un rol diferent al seu.

Inicialment, el conjunt de rutes de cada rol d'usuari es va guardar en un fitxer JavaScript a l'aplicació client. No obstant això, es va veure que fer-ho d'aquesta manera suposava una vulnerabilitat, ja que qualsevol pot veure i modificar aquests fitxers. Així doncs, es va decidir optar per una opció més segura: guardar aquesta informació al servidor i fer-li una petició d'accés cada vegada que un usuari vulgui accedir a una altra ruta. Concretament, es va habilitar la crida `GET /api/auth/access/:nom-ruta`, que fa servir el *token* d'autenticació de l'usuari per obtenir-ne el rol.

9.1.3 Models de dades

En aquesta subsecció, es detallarà com es defineixen els models de dades que, mitjançant l'ORM `Sequelize` (veure capítol 7), permeten mantenir les taules de la base de dades.

Al principi de l'execució de l'aplicació servidor, es fa servir Sequelize per establir una connexió amb la base de dades i sincronitzar-hi els models de dades definits. Cadascun dels models s'ha declarat en un fitxer diferent en què se n'indiquen els atributs i les relacions, entre altres paràmetres i configuracions.

A tall d'exemple, a la figura 9.7 es pot veure com està definit el model de dades per als estudis. Més concretament, es defineix mitjançant el mètode “define”, que rep tres paràmetres: nom del model en singular, atributs i configuracions addicionals.

El nom del model s'indica en singular, però el nom que pren la taula corresponent s'expressa, tal com es sol fer, en plural. En quant als atributs, cadascun accepta un conjunt de paràmetres determinat, com ara el tipus, si és clau primària, si pot ser nul, etc.

L'opció “paranoid” del bloc de configuracions addicionals indica que l'eliminació d'un estudi no sigui definitiva, sinó que, en comptes d'això, se li assigni la data d'eliminació.

A més, també es pot veure com s'obtenen els noms dels estudis, els quals no apareixen als fitxers dels plans docents (veure capítol 4). Sequelize permet definir atributs de tipus “VIRTUAL” que, en comptes d'emmagatzemar-se a la base de dades, es calculen mitjançant el respectiu mètode “get”. En aquest cas, els noms dels estudis estan mapejats amb l'abreviació corresponent en un fitxer JSON allotjat al servidor. Sens dubte, no és la millor solució; l'opció òptima seria obtenir totes aquestes dades de les bases de dades de la universitat (veure capítol 12).

```
const Study = sequelize.define(
  'Study',
  {
    abv: {
      type: DataTypes.STRING(8),
      primaryKey: true
    },
    name: {
      type: DataTypes.VIRTUAL,
      get() {
        return hardDataUtil.getValue('studies', this.abv) || null;
      }
    }
  },
  {
    paranoid: true,
```

Figura 9.7: Fragment de la definició del model de dades per als estudis.

Finalment, Sequelize proporciona un conjunt de mètodes que permet definir les relacions entre els models. No obstant això, presenten un inconvenient: necessiten que tots els models implicats en les relacions estiguin declarats. Per aquest motiu, s’ha hagut d’encapsular el codi corresponent de cada model dins de funcions que es criden un cop s’han importat tots. Aquestes funcions reben tots els models que necessiten i els fan servir per definir les relacions.

A la figura 9.8 es pot veure com es defineixen les relacions del model de dades per als estudis. Les crides al mètode “belongsTo” indiquen que un estudi pertany a un altre model i estableixen el nom de la clau forana de la relació. En aquests casos, la clau forana forma part de la taula dels estudis.

D’altra banda, el mètode “belongsToMany” serveix per definir relacions “molts a molts”. Se li ha d’indicar el model que faci de taula intermitja i els noms de les respectives claus foranes.

En darrer lloc, el mètode “hasMany” indica que un estudi pertany a múltiples instàncies d’un altre model. També estableix el nom de la clau forana que, en aquest cas, forma part de la taula de l’altre model.

```
Study.associate = ({ User, School, Subject, StudySubject, Group, StudyGroup, GenericTimeBlock }) => {
  Study.belongsTo(User, { foreignKey: 'coordinador' });

  Study.belongsTo(School, { foreignKey: 'school' });

  Study.belongsToMany(Subject, {
    through: StudySubject,
    foreignKey: 'study',
    otherKey: 'subject'
  });
  Study.hasMany(StudySubject, { foreignKey: 'study' });

  Study.belongsToMany(Group, {
    through: StudyGroup,
    foreignKey: 'study',
    otherKey: 'group'
  });
  Study.hasMany(StudyGroup, { foreignKey: 'study' });

  Study.hasMany(GenericTimeBlock, { foreignKey: 'study' });
};
```

Figura 9.8: Funció que defineix les relacions del model de dades per als estudis.

9.1.4 Enviament de correus electrònics

En aquesta subsecció, es parlarà sobre l’estat de l’enviament de correus electrònics.

Tal com s’ha vist en altres capítols, l’aplicació servidor ha d’enviar correus electrònics tant per a l’activació d’usuaris com per al restabliment de contrasenyes.

Després d'implementar-ho, es van provar múltiples servidors que efectuessin els enviaments, però, malauradament, no es va aconseguir fer-ho funcionar.

De moment, en comptes de ser enviat, el contingut dels correus electrònics es mostra a la consola per tal d'obtenir-ne les dades necessàries per fer funcionar l'aplicació. S'ha decidit abordar el tema del servidor d'enviament un cop s'hagi de llançar l'aplicació a producció, en un possible futur.

9.1.5 Implementació de l'API

En aquesta subsecció, es donaran a conèixer alguns detalls sobre el disseny i la implementació de l'API.

Un dels aspectes més complicats de tractar ha estat la recuperació eficient de les dades que el client requereix en cada moment. En la majoria de rutes i vistes del client, la informació que es necessita involucra bastantes taules de la base de dades, moltes de les quals estan relacionades entre si. Això comporta haver de fer un nombre considerable de crides al servidor, cosa que penalitza notablement el rendiment. A més a més, en molts casos, només es requereix una part concreta dels atributs de les entitats.

Degut a aquests motius, s'ha implementat un conjunt de mecanismes a l'API que permeten fer peticions més potents i eficients. Al conjunt de crides corresponent de cada taula, s'hi ha afegit `POST /api/<nom-entitat>/filter`, la qual conté les parts següents:

- **Paràmetres de la URL**

- `fields`: llista del nom dels atributs que es volen recuperar de la taula en qüestió. El servidor s'encarrega de validar que aquests atributs concordin amb els atributs de la taula i que no estiguin restringits; també s'ha implementat la possibilitat de restringir atributs i que no es puguin recuperar, com és el cas de la contrasenya dels usuaris. Per exemple, si només es vol recuperar el codi i el nom de les assignatures, la petició seria:

```
POST /api/subjects/filter?fields=code,name
```

- `include`: llista del nom de les taules que es volen incloure a la resposta, les quals han d'estar relacionades amb la taula en qüestió. Per exemple, si es vol obtenir un conjunt d'assignatures i incloure els estudis als quals pertany cadascuna, la petició seria:

```
POST /api/subjects/filter?include=Study
```

A diferència del paràmetre `fields`, `include` té una implementació dedicada per a cada taula, ja que la manera d'obtenir les relacions pot variar. Per tal de validar aquest paràmetre, per a cada taula es pot definir un conjunt d'inclúdes permesos.

- **Cos de la petició**

- `data`: conjunt de valors de filtratge sobre la taula en qüestió. Permet indicar filtres d'un o múltiples valors de cadascun dels atributs. A més, també es poden definir restriccions. Per exemple, si només es volen obtenir les assignatures de sis o nou crèdits que es cursin durant el segon semestre, el cos de la petició seria:

```
{ data: { credits: [6, 9], semester: 2} }
```

- `associations`: conjunt de valors de filtratge sobre la clau primària de les taules relacionades amb la taula en qüestió. Permet indicar filtres segons les relacions. A més, també es poden definir restriccions. Per exemple, si només es volen obtenir les assignatures de GEINF i GDDV, el cos de la petició seria:

```
{ associations: { study: ["GEINF", "GDDV"] } }
```

Tots aquests mecanismes es poden combinar per tal de formar peticions complexes que milloren el rendiment de l'aplicació.

En darrer lloc, cal comentar que s'ha implementat un sistema que evita que el processament d'una petició es quedi penjat davant d'errors inesperats. La solució es basa en encapsular els controladors de les peticions amb una funció que els crida dins d'un bloc *try-catch* i, en cas d'error, crida la següent funció de la línia de processament de peticions. Concretament, aquesta funció escriu l'error als *logs* (veure subsecció 9.1.1) i envia una resposta 500 (*internal server error*) al client. Cal remarcar que el motiu dels errors no s'envia per tal d'evitar-ne usos mal intencionats.

9.1.6 Proves de la base de dades

En aquesta subsecció, s'explicarà com s'han realitzat les proves a la base de dades i se n'aportarà algun exemple.

Les proves de la base de dades han consistit, principalment, en comprovar que l'aplicació servidor hi crea les taules correctament, així com els atributs, les claus foranes, etc. L'eina utilitzada per fer-les ha estat `DataGrip` (veure capítol 7).

Primerament, s'ha comprovat que s'hagin generat totes les taules corresponents als models de dades definits a l'aplicació servidor (veure subsecció 9.1.3). A la

figura 9.9 se'n pot veure el conjunt (noms en anglès).

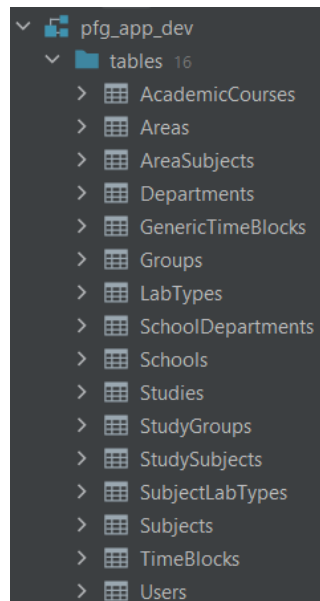


Figura 9.9: Taules de la base de dades generades per l'aplicació servidor.

Després d'haver verificat la creació de les taules, s'ha comprovat que els atributs, les restriccions i les claus de cadascuna siguin correctes. Com a exemple, a la figura 9.10 es pot veure el resultat de la prova realitzada a la taula d'estudis (*Studies*). Les claus principals es marquen amb una clau groga, mentre que les foranes es marquen amb una blava. La resta d'atributs s'indiquen amb gris i el cercle inferior esquerre d'alguns significa que tenen una restricció "NOT NULL", és a dir, que mai pot ser nul.

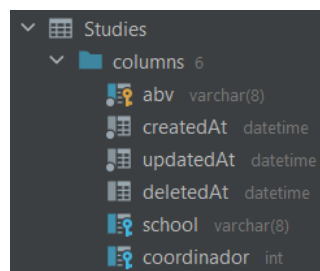


Figura 9.10: Atributs de la taula *Studies*.

9.1.7 Proves de l'API

En aquesta subsecció, s'explicarà com s'han realitzat les proves a l'API i se n'aportarà algun exemple.

Aquesta tipologia de proves s'ha dut a terme per verificar el processament de peticions que efectua l'aplicació servidor. L'eina utilitzada per fer-les ha estat Postman (veure capítol 7), el qual ha permès llançar peticions HTTP contra l'API i visualitzar-ne les respostes.

En primer lloc, s'ha validat el funcionament de l'autorització a les diferents crides (veure subsecció 9.1.2). Com que el mecanisme d'autorització és el mateix per a totes, amb una comprovació és suficient. A la figura 9.11 es pot veure com l'aplicació servidor no deixa efectuar la petició GET /api/studies/GEINF, ja que requereix l'existència d'un *token* al camp "Authorization" de la capçalera.

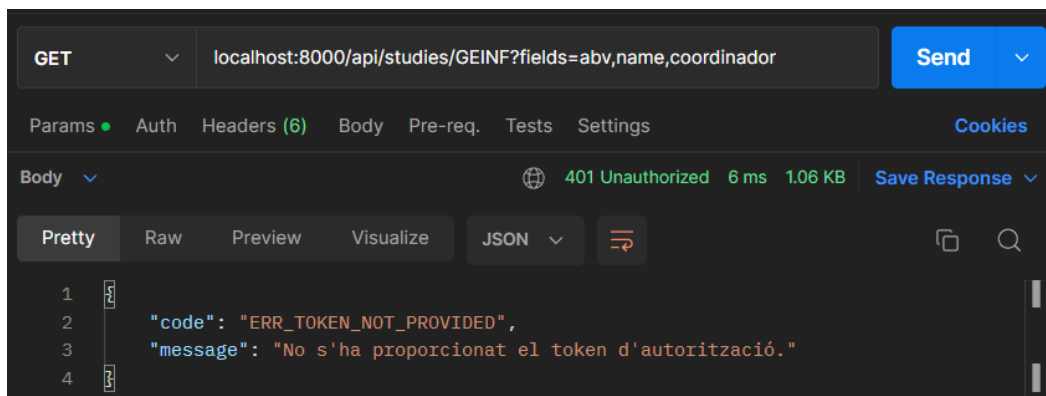


Figura 9.11: Resposta del servidor quan una petició no duu el *token* per a l'autorització.

D'altra banda, a la figura 9.12 es pot veure com, si el *token* és invàlid, tampoc deixa efectuar la petició anterior.

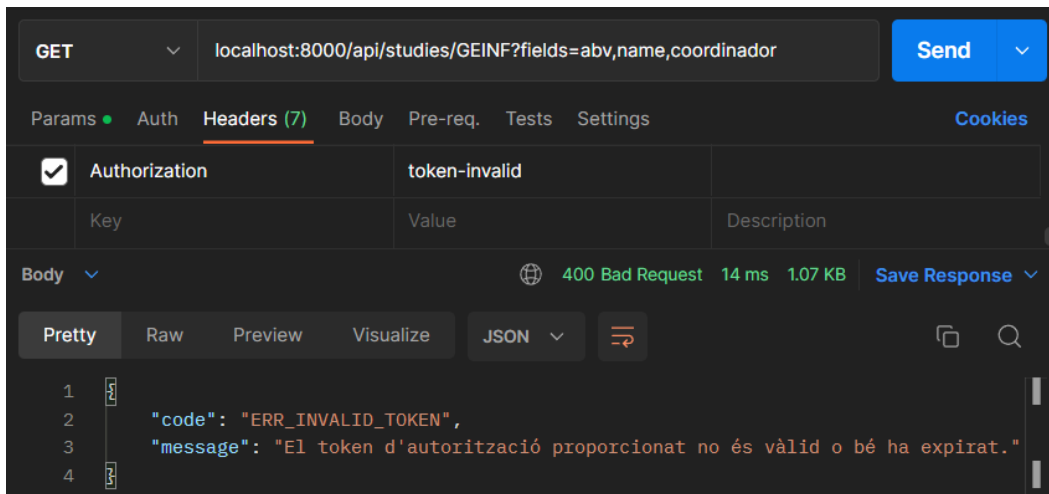


Figura 9.12: Resposta del servidor quan el *token* per a l'autorització d'una petició és invàlid.

També cal afegir que s'han realitzat proves per validar que els usuaris no puguin crear-ne ni esborrar-ne d'altres sense disposar dels permisos corresponents. Per exemple, un usuari que no sigui Administrador no pot crear ni esborrar usuaris Coordinadors. A la figura 9.13 es pot veure una prova en què un Director de departament intenta eliminar un Administrador.

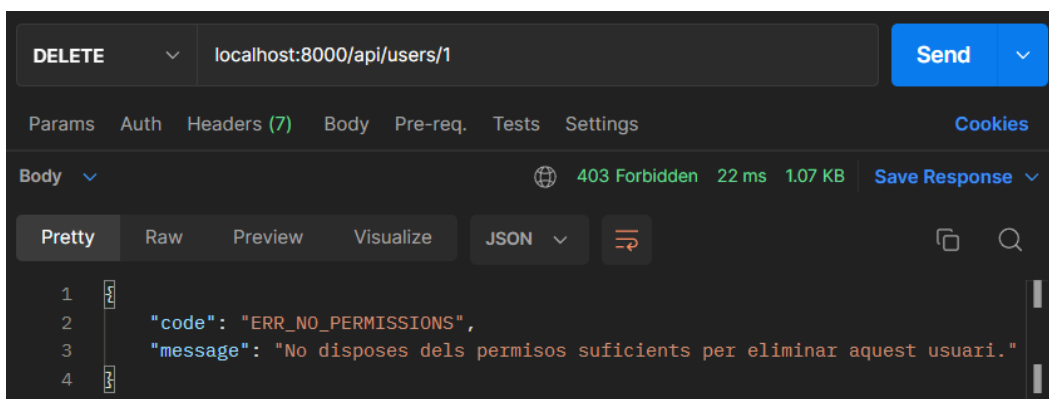


Figura 9.13: Resposta del servidor quan un usuari intenta esborrar-ne un altre i no té els permisos suficients.

En darrer lloc, es mostrarà el resultat d'una de les proves que comproven el funcionament de les crides de filtratge descrites a la subsecció 9.1.5. A la figura 9.14 es pot veure l'exemple d'una petició de filtratge d'assignatures. Concretament, li indica al servidor el següent:

- **fields**: només recupera els atributs “code” (codi) i “name” (nom) de les assignatures.
- **include**: inclou també, per a cada assignatura, els “Study” (estudis) i els “Group” (grups) amb què està relacionada. Per facilitar certes funcionalitats, en aquest cas, incloure els grups també implica incloure’n els blocs horaris i els estudis amb què estan relacionats.
- **data**: recupera només les assignatures l’atribut “semester” (quadrimestre) de les quals sigui “1”.
- **associations**: recupera només les assignatures que estiguin relacionades amb almenys un dels “labType” (tipus de laboratori) anomenats “INFOR” o “LINUX”.



```
1  {
2    ... "data": {
3      ... "semester": 1
4    },
5    ... "associations": {
6      ... "labType": ["INFOR", "LINUX"]
7    }
8  }
```

Figura 9.14: Exemple d’una petició de filtratge d’assignatures.

La resposta que retorna l’aplicació servidor davant de la petició anterior es pot veure a la figura 9.15. Degut a la grandària de la resposta, només se’n mostra un fragment que conté una part de les dades d’una de les assignatures retornades. Es pot comprovar que només en retorna els atributs sol·licitats i que inclou els estudis i els grups tal com s’havia demanat. Pel que fa al filtratge, encara que no es pugui observar a la figura, només ha retornat les assignatures que compleixen els filtres indicats.

```
{
  "code": "3105G07012",
  "name": "Multimèdia i interfícies d'usuari",
  "LabTypes": [
    {
      "name": "INFOR"
    }
  ],
  "Studies": [
    {
      "name": "Enginyeria Informàtica",
      "abv": "GEINF",
      "StudySubject": {
        "course": 3
      }
    },
    {
      "name": "Disseny i Desenvolupament de Videojocs",
      "abv": "GDDV",
      "StudySubject": {
        "course": 2
      }
    }
  ],
  "Groups": [
    {
      "id": 1791,
      "type": "small",
      "number": 1,
      "TimeBlocks": [
        {
          "start": null,
          "id": 1791,
          "day": null,
          "duration": 120,
          "week": null
        }
      ]
    }
  ],
  "Studies": []
},
```

Figura 9.15: Fragment de la resposta de l'aplicació servidor davant d'una petició de filtratge d'assignatures.

9.2 Aplicació client

9.2.1 Enrutament

En aquesta subsecció, s'indicaran alguns dels punts més rellevants i problemàtics pel que fa a l'enrutament de l'aplicació client.

En primer lloc, s'ha implementat un mecanisme que assegura que l'usuari no pugui accedir a rutes que no existeixen ni a rutes a les quals no pot fer-ho. Aquest mecanisme consisteix en l'execució d'una funció que s'executa abans d'entrar a qualsevol ruta. A la figura 9.16 es pot veure exactament com és aquesta funció, funcionament de la qual s'explica a continuació:

1. S'assegura que les dades i l'estat d'autenticació de l'usuari estiguin sincronitzats.
2. Si l'usuari està autenticat, es continua amb el procés. En cas contrari, es comprova que la ruta a la qual vol accedir no requereixi autenticació. Si no en requereix, se li permet l'accés, sinó se'l redirigeix a la ruta de *login* (autenticació).
3. En aquest punt es sap que l'usuari està autenticat. Tot seguit, es comprova que la ruta existeixi i que, a més a més, l'usuari tingui permís per accedir-hi (veure subsecció 9.1.2). Si la ruta existeix i l'usuari hi té accés, se li permet entrar-hi. En cas contrari, se'l redirigeix a la ruta per defecte del seu rol d'usuari.

```
router.beforeEach(async to => { // to: ruta a accedir.
  const authStore = useAuthStore();
  await authStore.refreshing; // Assegurar sincronització.
  if (authStore.isLoggedIn) { // Usuari autenticat?
    try {
      // La ruta existeix i l'usuari hi té accés?
      if (to.name && (await authStore.hasAccessTo(to.name))) {
        return; // Permís concedit.
      }
    } catch (_e) {}
    return { name: authStore.defaultView }; // Redirecció a ruta per defecte.
  }
  if (to.meta.noAuth) { // La ruta requereix autenticació?
    return; // Permís concedit.
  }
  return { name: 'login' }; // Redirecció a ruta d'autenticació.
});
```

Figura 9.16: Funció que s'executa abans d'entrar a qualsevol ruta.

D'altra banda, la ruta més complexa d'implementar ha estat la de veure o modificar horaris d'estudis. S'ha volgut fer que tots els horaris siguin accessibles directament des d'una URL, per si en un futur cal fer-hi redireccions, per si un usuari se'n vol guardar un als marcadors del navegador, etc. Més concretament, la ruta adopta la forma següent:

- `/horaris-graus`: s'hi mostra la interfície que permet escollir l'horari a visualitzar o a modificar.
- `/horaris-graus/<estudi>/<curs>/<semestre>?action=<view|edit>`: s'hi mostra la interfície que permet visualitzar o modificar un horari concret.

A la figura 9.17 es pot veure un fragment de la definició d'aquesta ruta. Concretament, es pot veure com es defineix la ruta general i les rutes filles (*children*).

- **Ruta general (“horarisGraus”)**: ruta el component de la qual conté el component de visualització de les rutes filles. La seva funció és redirigir aquest component a la ruta filla “studyScheduleChoosing”.
- **Ruta filla d'elecció d'horaris (“studyScheduleChoosing”)**: ruta que mostra la interfície d'elecció de la visualització o modificació d'horaris d'estudis.
- **Ruta filla de URL invàlida (“studyScheduleInvalidMatch”)**: ruta que captura totes les URL invàlides de rutes filles i fa la redirecció a (“studyScheduleChoosing”).
- **Ruta filla de visualització o modificació d'horaris (“studySchedule”)**: ruta que mostra la interfície de visualització o modificació de l'horari de l'estudi corresponent. Concretament, utilitza una expressió regular que defineix el format correcte de la URL, ja que s'hi ha d'indicar una abreviació d'estudi, un curs, un quadrimestre i l'acció a realitzar. Per exemple, si es vol modificar l'horari del primer quadrimestre del tercer curs de GEINF, la URL de la ruta seria:

```
/horaris-graus/GEINF/3/1?action=edit
```

```
{
  path: '/horaris-graus',
  name: 'horarisGraus',
  component: () => import('@views/HorarisGrausView.vue'),
  redirect: { name: 'studyScheduleChoosing' },
  children: [
    {
      path: '',
      name: 'studyScheduleChoosing',
      component: () => import('@views/scheduling/StudyChoosing.vue')
    },
    {
      path: ':invalidMatch(.*)*',
      name: 'studyScheduleInvalidMatch',
      redirect: { name: 'studyScheduleChoosing' }
    },
    {
      path: ':abv([a-zA-Z]+)/:course(\\d+)/:semester(1|2)',
      name: 'studySchedule',
      component: () => import('@views/scheduling/StudySchedule.vue'),
    }
  ]
}
```

Figura 9.17: Fragment de la definició de la ruta de visualització o modificació d'horaris d'estudis.

Tal com es pot intuir, la validació del format de la URL de la ruta “studySchedule” no és suficient per evitar errors i vulneracions de seguretat. Concretament, es poden donar les situacions següents:

- L'estudi o el curs indicat no existeix.
- L'acció és visualitzar i l'usuari no té permís per veure els horaris de l'estudi indicat.
- L'acció és modificar i l'usuari no té permís per fer-ho amb els horaris de l'estudi indicat.
- L'acció no és ni visualitzar ni modificar.

Pel que fa a les dues primeres, la solució implementada ha estat el redireccionament de l'usuari a la ruta d'elecció d'horaris (“studyScheduleChoosing”). Pel que fa a les últimes, la solució ha estat comprovar si l'usuari disposa de permisos per visualitzar els horaris de l'estudi indicat. En cas afirmatiu, l'acció es canvia a visualitzar. Si no, també se'l redirecciona a la ruta d'elecció d'horaris.

9.2.2 Solapaments de tipus de laboratori

En aquesta subsecció, es repassaran els aspectes més importants de la implementació de la detecció dels solapaments de tipus de laboratori.

Els solapaments de tipus de laboratori es produeixen quan s'excedeix la quantitat d'aules d'un mateix tipus de laboratori durant una franja horaria determinada. Cal notar que, en aquesta classe de solapaments, hi afecten els horaris d'un quadrimestre concret de tots els cursos de tots els estudis.

El primer aspecte a tenir en compte i a decidir és com fer la divisió de cada dia de l'horari en franges. Primerament, es van plantejar diverses solucions que es basaven en dividir els dies en intervals fixes, com ara de 8:00h a 8:30h, de 9:00h a 9:30h, etc.

Encara que podrien haver funcionat, aquestes solucions no permetien fer una detecció del tot exacta: es podria donar el cas, per exemple, que un tipus de laboratori no estigués disponible de les 8:30h a les 8:45h, però sí de 8:45h a 9:00h. Amb una divisió de les franges en intervals de mitja hora, el període de les 8:45h a les 9:00h es consideraria no disponible, quan en realitat sí que ho està.

Per aquest motiu, es va optar per fer una divisió dels dies en franges variables en funció de tots els blocs horaris implicats. D'aquesta manera, es pot fer una detecció exacta del nombre d'aules de cada tipus de laboratori ocupades en cada moment de l'horari.

A la figura 9.18 es pot veure un exemple de com es faria la divisió en franges d'un dilluns a partir dels blocs horaris pertanyents a grups petits que s'imparteixen al tipus de laboratori "INFOR", que disposa de deu aules. Per fer les delimitacions, es tenen en compte les hores d'inici i de fi de cada bloc horari. Una franja correspon al període comprès entre dos inicis i/o fins consecutius de blocs horaris.

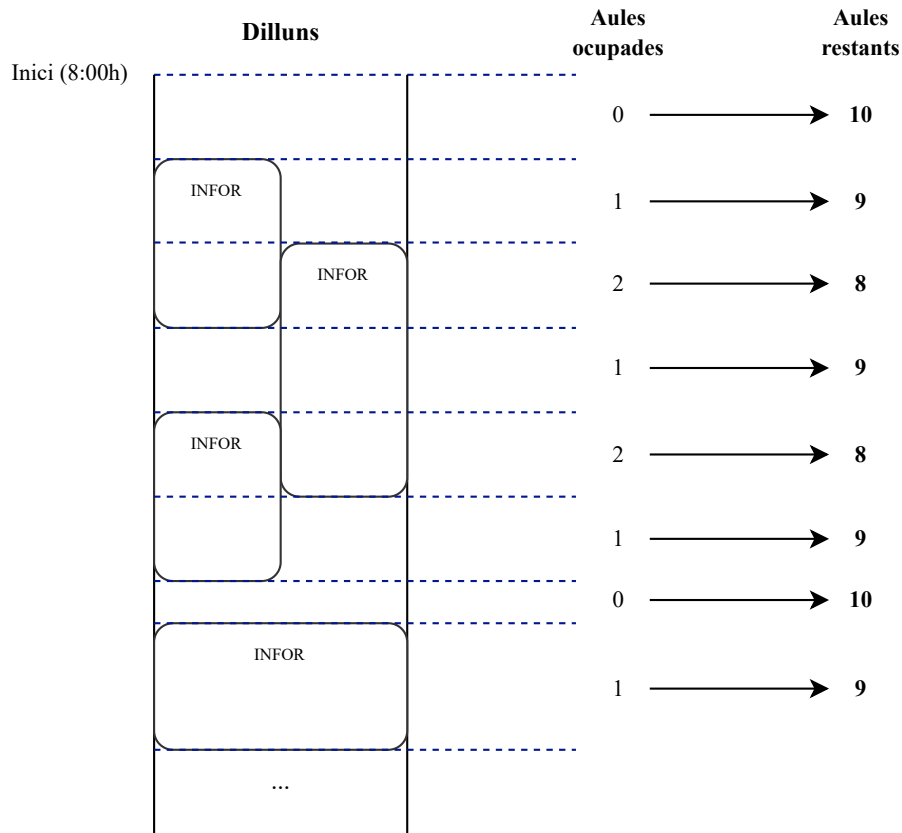


Figura 9.18: Exemple de la divisió d'un dia en franges per a calcular l'ocupació de les aules del tipus de laboratori "INFOR".

Tal com s'ha indicat en altres capítols, els solapaments s'han de calcular mentre s'està modificant un horari concret. Per aquest motiu, en aquestes situacions, només cal calcular l'ocupació dels tipus de laboratori implicats. A més, també s'han de tenir en compte les setmanes en què els blocs horaris estan situats. La solució implementada es basa en les dues fases següents:

- Quan es carrega la interfície de modificació d'un horari: calcular l'ocupació de les aules dels tipus de laboratori implicats, per possibilitar-ne una futura consulta ràpida. Aquesta informació es torna a calcular cada vegada que l'horari pateix una modificació.
- Quan es volen consultar els solapaments dels tipus de laboratori en què s'imparteix el grup d'un bloc horari determinat: consultar-ne l'ocupació per tal de saber tant si s'estan solapant com en quines franges provocarien solapament en cas que s'hi col·loquessin.

A continuació, s'explicarà com s'han aplicat, en aquest cas, els mecanismes de l'API detallats a la secció 9.1. En primer lloc, cal recuperar els blocs horaris dels grups de totes les assignatures implicades, és a dir, de les assignatures que tinguin assignats almenys un dels tipus de laboratori que apareixen a l'horari en qüestió. Per fer-ho, es fa servir una sola crida al servidor, gràcies al disseny i implementació de l'API.

A la figura 9.19 es pot veure com es realitza aquesta petició amb exactitud. Concretament, s'estan filtrant les assignatures segons quadrimestre (*filterData*) i conjunt de tipus de laboratori (*associations*). A més, pel que fa als atributs, només se'n recuperen el codi i el nom (*fields*): els únics necessaris. A més, per a cadascuna, també se n'obtenen els grups i blocs horaris (*include*). D'altra banda, conèixer-ne els estudis és essencial per poder donar informació sobre els cursos i estudis implicats en els possibles solapaments que puguin tenir lloc.

```
async loadImplicatedSubjects() {
  const { data: subjects } = await subjectsApi.list({
    params: { fields: 'code,name', include: 'Study,Group' },
    filterData: { semester: this.semester },
    associations: { labType: this.labTypeNames }
  });

  return subjects;
},
```

Figura 9.19: Petició per recuperar la informació necessària per calcular solapaments de tipus de laboratori.

A partir d'aquest punt, es calcula l'ocupació de cada tipus de laboratori per a cada dia i setmana. D'aquesta manera, quan l'usuari necessiti veure els solapaments d'un conjunt de tipus de laboratori concret, l'aplicació només haurà de consultar les dades d'ocupació corresponents.

9.2.3 Usabilitat i estètica

En aquesta subsecció, es donaran a conèixer alguns dels aspectes més importants pel que fa a la usabilitat i a l'estètica de la part de les interfícies d'horaris.

Una de les parts més rellevants i complexes en quant a estètica i llegibilitat ha estat la distribució i la mida dels blocs horaris. S'han implementat algorismes que, per a cada dia, calculen tant la posició horitzontal com l'amplada dels blocs horaris, en funció de les col·lisions que es puguin produir amb la resta.

Per a cada dia, primerament, es divideixen els blocs horaris en "columnes" que formen "grups": cada columna conté blocs horaris que no col·lisionen entre ells, mentre que cada grup conté un conjunt de columnes que col·lisionen entre elles. D'aquesta manera, es poden posicionar els blocs horaris grup per grup amb la certesa que no col·lisionen. A la figura 9.20 es pot veure la implementació d'aquest algorisme. Per tal que la figura no quedi massa llarga, s'ha plegat la part d'ordenació, que simplement ordena els blocs horaris en funció de les hores d'inici i de fi (de més aviat a més tard).

Dins de cada grup, la posició horitzontal relativa dels blocs horaris de les columnes és l'índex de la columna a què pertanyen respecte el total de columnes del grup. D'altra banda, per calcular-ne l'amplada cal veure, a partir de l'índex de la seva columna, quantes columnes d'aquell grup hi ha (a la dreta) amb què no col·lisiona amb cap dels blocs. Gràcies a aquest nombre i al nombre de columnes del grup en qüestió, es pot calcular l'amplada relativa dels blocs horaris.

```

const layoutTimeBlocks = timeBlocks => {
  const groups = []; // Grups
  let columns = []; // Conjunt de columnes que forma un grup
  let lastTimeBlockEndMinutes = 0; // Hora (en minuts) de fi del bloc que
  // acaba més tard del grup que s'està formant
  timeBlocks
    .sort((tb1, tb2) => {...
    }) // Ordenació de més aviat a més tard
    .forEach(timeBlock => {
      if (
        lastTimeBlockEndMinutes &&
        timeToMinutes(timeBlock.start) >= lastTimeBlockEndMinutes
      ) {
        // L'hora de fi del bloc que acaba més tard del grup és més petita
        // que la d'inici del següent bloc a processar: el grup ja està
        // complet i se'n crea un de nou
        groups.push(columns);
        columns = [];
        lastTimeBlockEndMinutes = 0;
      }
      // Busca una columna l'últim bloc de la qual no col·lisió amb el
      // que s'està processant
      const col = columns.find(col => !collide(col.at(-1), timeBlock));
      if (col) {
        // Si es troba, significa que el bloc que s'està processant no
        // col·lisió amb cap dels d'aquella columna (per l'ordenació)
        col.push(timeBlock);
      } else {
        // Si no es troba, significa que sí que col·lisió amb els blocs
        // de la columna i, per tant, ha de pertànyer a una de nova
        columns.push([timeBlock]);
      }
      const endMinutes = getEndMinutes(timeBlock.start, timeBlock.duration);
      if (lastTimeBlockEndMinutes <= endMinutes) {
        // El bloc que s'estava processant acaba més tard que l'últim del grup
        lastTimeBlockEndMinutes = endMinutes;
      }
    });
  // Concatena la llista de grups amb l'últim grup calculat
  return [...groups, columns];
};

```

Figura 9.20: Algoritme que divideix els blocs horaris d'un dia concret en grups i columnes.

A la figura 9.21 es pot veure com es calcula el nombre de columnes màxim que pot ocupar horitzontalment un bloc.

```
const getTimeBlockColSpan = (timeBlock, colIndex, group) => {
  let colSpan = 1; // Multiplicador de l'amplada relativa del
  // bloc

  // A partir de la següent columna, es busca si el bloc
  // col·lisiona amb alguna de les restants. Mentre no
  // ho faci, significa que el bloc pot ocupar-ne
  // l'espai corresponent i s'incrementa el multiplicador.
  // En canvi, quan es detecta una col·lisió, significa
  // que ja té l'amplada màxima que pot tenir
  for (const col of group.slice(colIndex + 1)) {
    if (col.some(timeBlock2 => collide(timeBlock, timeBlock2))) {
      break;
    }
    colSpan++;
  }

  return colSpan;
};
```

Figura 9.21: Algoritme que calcula el nombre de columnes que pot ocupar horitzontalment un bloc horari.

També és important destacar que, per tal de millorar considerablement la fluïdesa tant de l'arrossegament, com de l'allargament i l'escurçament dels blocs horaris, s'ha fet el següent: quan es deixa anar un bloc horari després d'haver-lo arrossegat, en primer lloc, s'executen els algoritmes de col·locació per tal d'efectuar ràpidament el canvi visual. Un cop s'ha col·locat, es fa la crida corresponent a l'API i s'espera la resposta. Si ha anat bé, ja no cal fer res més (a part de notificar-ho a l'usuari). En canvi, si ha anat malament, es desfà el canvi visual i se'n notifica a l'usuari. Gràcies a això, com que en la gran majoria de casos no es produeixen errors, és molt millor, en termes de fluïdesa, efectuar el canvi visual sense esperar la resposta del servidor.

A més a més, pel que fa al text que es mostra als blocs horaris, quan l'àrea és relativament petita, a partir d'un cert punt se n'abreuen els textos per tal que no sobresurtin. L'algoritme d'abreviació té en compte els textos d'una sola paraula ("Estadística" → "Est") i els números que hi puguin aparèixer ("Metodologia i Tecnologia de la Programació II" → "MTP2").

D'altra banda, es prohibeix tant l'arrossegament com l'allargament d'un bloc horari fora dels límits permesos. A més a més, el canvi manual de l'hora d'inici, de fi i la duració és assistit:

- Quan es modifica l'hora d'inici, es modifica, consegüentment, l'hora de fi. No es deixa introduir una hora d'inici que, sumada amb la duració, faci

que l'hora de fi surti dels límits establerts.

- Quan es modifica l'hora de fi, es modifica, consegüentment, l'hora d'inici. No es deixa introduir una hora de fi que, si se li resta la duració, faci que l'hora d'inici surti dels límits establerts.
- Quan es modifica la duració, es modifica, consegüentment, l'hora de fi. Si amb aquesta duració, l'hora de fi es surt del límit, en comptes de permetre-ho, es fixa al límit i es modifica l'hora d'inici de manera corresponent.
- Si l'usuari, pel motiu que sigui, aconsegueix introduir valors invàlids, l'aplicació no li deixa guardar les modificacions.

En quant als colors de les interfícies d'horaris, s'ha escollit un color distintiu per a cada tipus de grup. A partir d'aquests colors, s'ha implementat un mecanisme que permet que l'estètica de qualsevol element relacionat amb un tipus de grup concret segueixi la mateixa línia de colors. A la figura 9.22 es pot veure com és la funció que rep un tipus de grup com a paràmetre i retorna una funció que en retorna el color específic d'un element en concret. Els noms d'aquests elements es defineixen al fitxer de constants (per exemple “botóGuardar”, “fonsBloc”, “fonsSetmana”, etc.).

```
const getStylingGetters = groupType => ({
  getColor: el => {
    const {
      timeBlockColorNames: colorNames,
      timeBlockColorTones: colorTones
    } = useConstants();
    return `${colorNames[groupType]}-${colorTones[el]}`;
  }
});
```

Figura 9.22: Funció que retorna la funció que permet obtenir el color d'un element concret segons un tipus de grup.

Un altre aspecte a destacar és que s'ha habilitat una manera per saber, directament sobre el calendari, l'hora d'inici, de fi i duració exactes de cada bloc horari. Es tracta d'una targeta que es desplega quan el ratolí es situa sobre d'un bloc concret. A la figura 9.23 se'n pot veure un exemple.

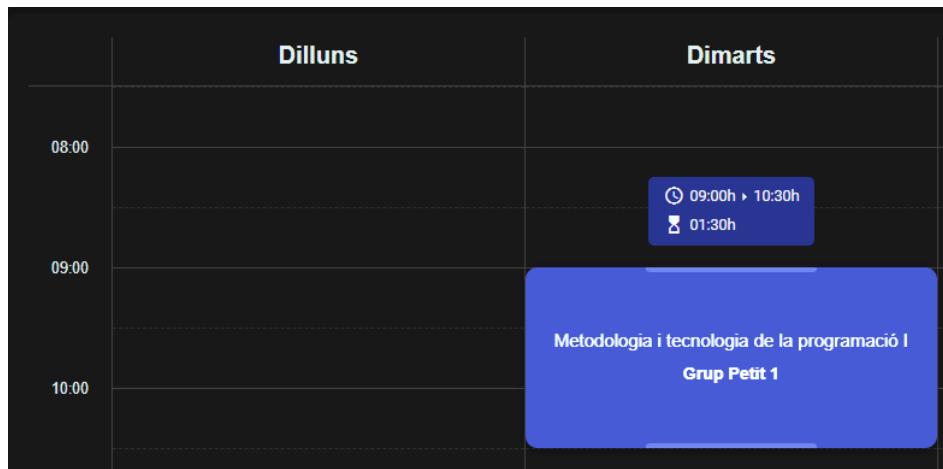


Figura 9.23: Targeta que indica l'hora d'inici, de fi i duració exactes d'un bloc horari.

Adicionalment, cal afegir que, després de contemplar diverses opcions per marcar els solapaments, s'ha optat per la que s'ha cregut més distingible i més atractiva estèticament: un quadre vermell pels blocs horaris solapats i una franja gris pels possibles solapaments que es marquen quan s'arrossega un bloc. A la figura 9.24 se'n pot veure un exemple.



Figura 9.24: Blocs horaris marcats com a solapats.

En darrer lloc, cal remarcar que s'han tingut en compte diferents mides de pantalla a l'hora d'implementar les interfícies: per tal d'adaptar-se a la pantalla, es modifiquen textos, canvia la disposició i mida de certs elements, etc.

Hi ha bastants més factors i mecanismes implementats que contribueixen a l'usabilitat i estètica de les interfícies de l'aplicació client. No obstant això, en aquesta subsecció s'han esmentat només les que s'han cregut més rellevants.

9.2.4 Proves

En aquesta subsecció, s'explicarà com s'han realitzat algunes de les proves a l'aplicació client i se n'aportaran exemples.

Pel que fa a la gestió d'usuaris, s'ha verificat que no aparegui un usuari ja assignat com a possible opció d'una altra assignació. Com a exemple, a la figura 9.25 es pot veure com, efectivament, l'usuari "Marta Fort" no apareix com a opció als desplegable d'assignació de Coordinadors a estudis, ja que ja està assignat.

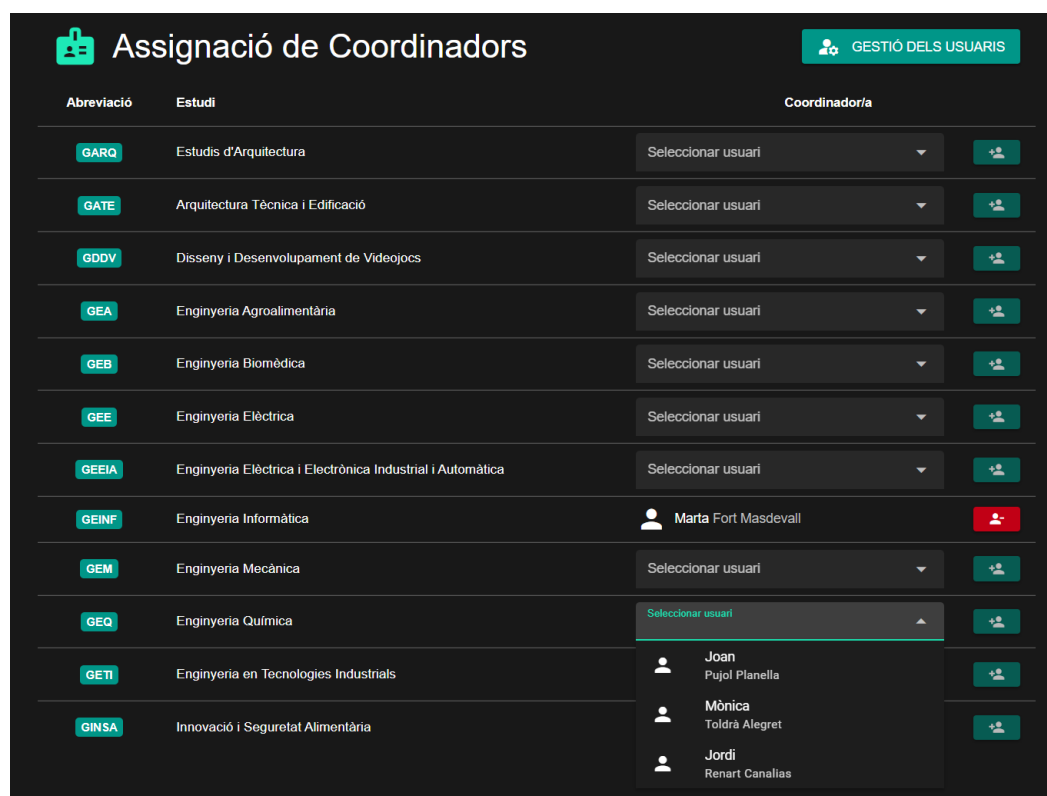


Figura 9.25: Contingut del desplegable d'usuaris quan ja n'hi ha algun d'assignat.

D'altra banda, també s'ha comprovat el funcionament dels algoritmes de col·locació de blocs horaris en situacions complexes (veure subsecció 9.2.3). A la figura 9.26 es pot veure un exemple de com es distribueixen els blocs horaris en una situació relativament complexa: no es tapen entre ells i la seva amplada és la màxima possible. A més, també s'hi pot apreciar que l'abreviació dels textos en funció de l'àrea funciona correctament.

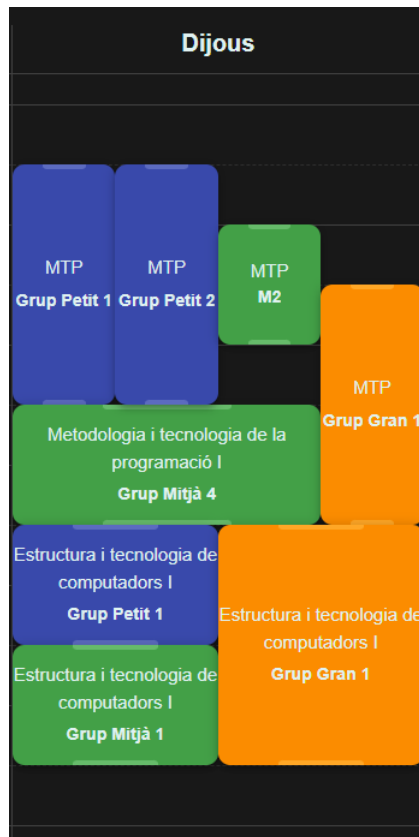


Figura 9.26: Distribució de blocs horaris al calendari en una situació complexa.

Una altra prova que s'ha considerat rellevant de comentar ha estat la del funcionament de les vistes setmanals. A les següents figures, es pot veure el mateix fragment d'un horari vist, respectivament, des de la vista general (veure figura 9.27), la vista de setmanes A (veure figura 9.28) i la vista de setmanes B (veure figura 9.29). A la general s'hi havien de mostrar tots els blocs, mentre que a la de setmanes A només s'hi havien de mostrar els blocs assignats a aquest tipus de setmana o a tots dos alhora. El mateix per a la vista de les B.

Les proves realitzades per verificar el funcionament dels diferents filtres de blocs horaris s'han realitzat de la mateixa manera i han resultat exitoses.



Figura 9.27: Fragment d'un horari vist des de la vista setmanal general.



Figura 9.28: Fragment d'un horari vist des de la vista de setmanes A.

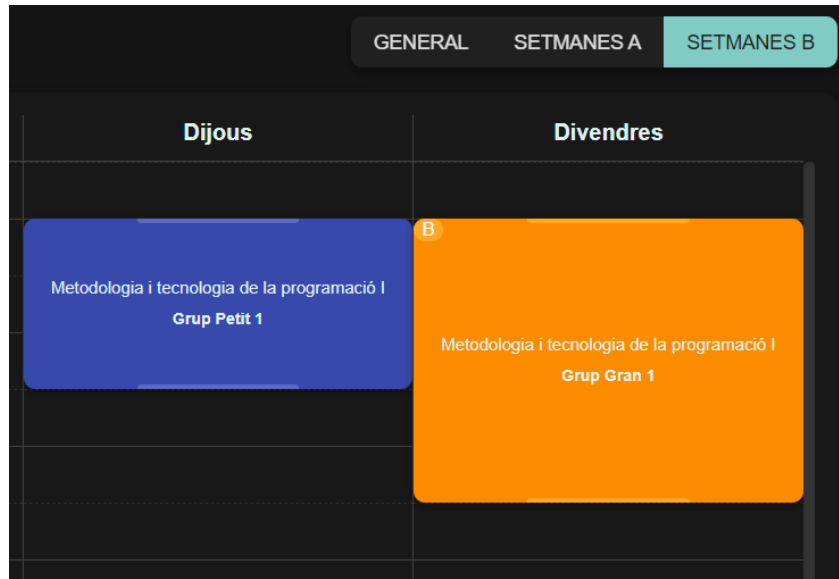


Figura 9.29: Fragment d'un horari vist des de la vista de setmanes B.

L'última prova important que es destacarà es basa en comprovar que els solapaments de tipus de laboratori es detectin correctament. Per a la prova, s'han situat blocs horaris al primer i tercer curs de GEINF i s'han assignat uns quants grups a GEINF i uns quants grups a GDDV, ja que s'han utilitzat assignatures compartides entre aquests dos estudis. A la figura 9.30 es pot veure com, efectivament, el tipus de laboratori "INFOR" està marcat com a solapat i a la dreta apareix la targeta indicant-ne els detalls, ja que s'hi ha situat el ratoí a sobre. En aquesta targeta s'hi mostren els cursos de cada estudi implicats en el solapament.

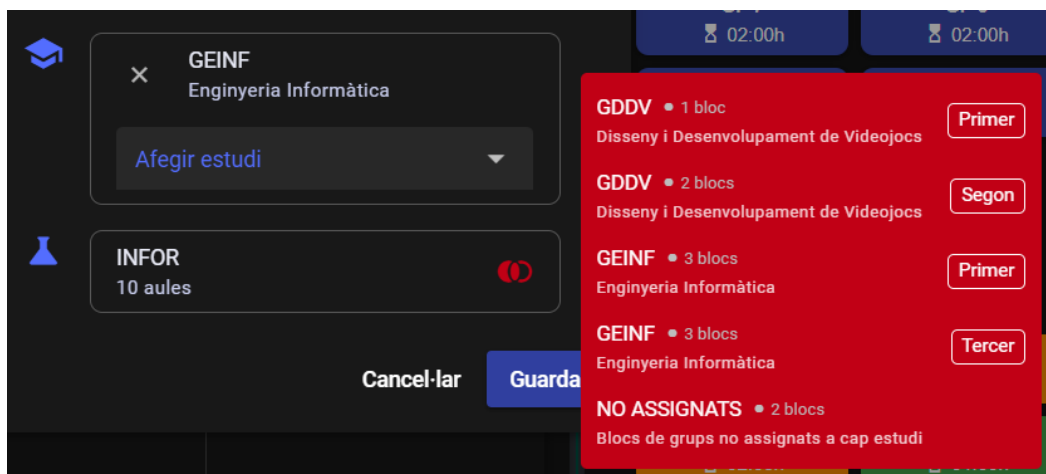


Figura 9.30: Detall dels solapaments d'un tipus de laboratori determinat.

Implantació i resultats

En aquest capítol, es presentaran els detalls sobre la implantació del projecte (veure secció 10.1) i els resultats obtinguts (veure secció 10.2).

10.1 Implantació en l'entorn local

En aquesta secció, es parlarà sobre com l'aplicació del projecte està implantada localment, tant el *front-end* i el *back-end* com el sistema gestor de bases de dades.

L'aplicació de *front-end* és operada pel servidor de desenvolupament que proporciona Vite (veure capítol 7). Aquest servidor està exposat al domini “localhost” i al port TCP 3000 de l'ordinador. Per tal que Vite serveixi l'aplicació, només cal accedir a l'adreça “http://localhost:3000” a través del navegador.

Al mateix temps, és necessari que l'aplicació de *back-end* també s'estigui executant i que pugui servir a les peticions que li faci el client. Aquesta aplicació utilitza Node.js (veure capítol 7) per muntar un servidor que exposa l'API a l'adreça “http://localhost:8000/api”. L'aplicació client coneix aquesta adreça i és capaç de llançar-hi peticions.

Per acabar, el sistema gestor de bases de dades també ha d'estar exposat a la xarxa local per tal que l'aplicació de *back-end* pugui comunicar-s'hi. Tal com s'ha vist al capítol 7, MySQL s'executa dins d'un contenidor de Docker, el qual n'exposa la connexió a través del port TCP 3306.

A continuació, es repassaran els passos que cal seguir per posar en marxa localment tot el conjunt de l'aplicatiu.

1. Instal·lar Docker [[Docker 2022](#)].
2. Obrir una terminal i descarregar una imatge de la versió més recent de MySQL:

```
> docker pull mysql:latest
```

3. Aixecar un contenidor amb la imatge descarregada:

```
> docker run --name=<nom contenidor> -p 3306:3306  
-e MYSQL_ROOT_PASSWORD=<contrasenya> -d mysql:latest
```

4. Obrir una terminal dins del contenidor:

```
> docker exec -it <nom contenidor> bash
```
5. Executar-hi MySQL i entrar la contrasenya escollida anteriorment:

```
> mysql -u root -p
```
6. Crear-hi l'usuari que farà servir l'aplicació de *back-end*:

```
> CREATE USER 'pfg-app-server'@'%'
IDENTIFIED BY 'pfg';
```
7. Proporcionar a l'usuari creat tots els privilegis de la base de dades:

```
> GRANT ALL PRIVILEGES ON * . *
TO 'pfg-app-server'@'%' ;
```
8. Fer efectius els canvis de permisos:

```
> FLUSH PRIVILEGES;
```
9. Crear la base de dades:

```
> CREATE SCHEMA pfg_app_dev;
```
10. Clonar el repositori de l'aplicació del projecte en una carpeta local, el qual està disponible a github.com/adriribas/pfg-application
11. Definir les variables d'entorn del servidor creant un fitxer anomenat “.env” (sense extensió) dins del directori “/packages/server” des de l'arrel del projecte que contingui les línies següents:
 - `HOST='localhost'`
 - `PORT=8000`
 - `DEBUG='pfgs:*`
 - `pfgs_authJwtPrivateKey='contrasenyaSecreta1'`
 - `pfgs_resetPasswordJwtPrivateKey='contrasenyaSecreta2'`
 - `pfgs_emailConfirmationJwtPrivateKey='contrasenyaSecreta3'`
 - `pfgs_user='pfg-app-server'`
 - `pfgs_dbSecret='pfg'`
12. Instal·lar la versió 16 de Node.js [[OpenJs Foundation 2022b](#)].
13. Obrir una terminal en el directori arrel de l'aplicació i instal·lar totes les dependències necessàries:

```
> npm install
```

14. Executar l'aplicació de *back-end*:

```
> npm run start:server
```
15. Obrir una altra terminal en el mateix directori i executar l'aplicació de *front-end*:

```
> npm run start:client
```
16. Accedir a <http://localhost:3000> des d'un navegador.

10.2 Resultats obtinguts

10.2.1 Consideracions inicials

En aquesta subsecció, es presentaran una sèrie de consideracions inicials que cal tenir en compte a la resta de la secció.

Els resultats es presentaran en forma de captures de pantalla que mostren el funcionament del programari desenvolupat durant el projecte. Totes les captures tindran la mateixa mida per tal d'oferir una perspectiva realística de com són les interfícies d'usuari.

A més a més, aquesta secció també fa la funció de manual d'usuari.

És important destacar que, per no sobrecarregar la memòria amb masses imatges, s'han obviat certes parts o estats de les interfícies que no són de vital importància, com ara notificacions d'informació o error, estats de càrrega, seleccionables desplecats, etc.

10.2.2 Autenticació

En aquesta subsecció, s'exposaran els resultats obtinguts pel que fa a l'autenticació.

A la figura 10.1 es pot veure la interfície resultant en la qual un usuari pot realitzar les accions següents:

- Entrar l'adreça de correu electrònic i contrasenya del seu usuari per tal d'autenticar-se.
- Prémer l'enllaç "Restableix-la aquí" per tal de procedir a restablir la contrasenya del seu usuari.

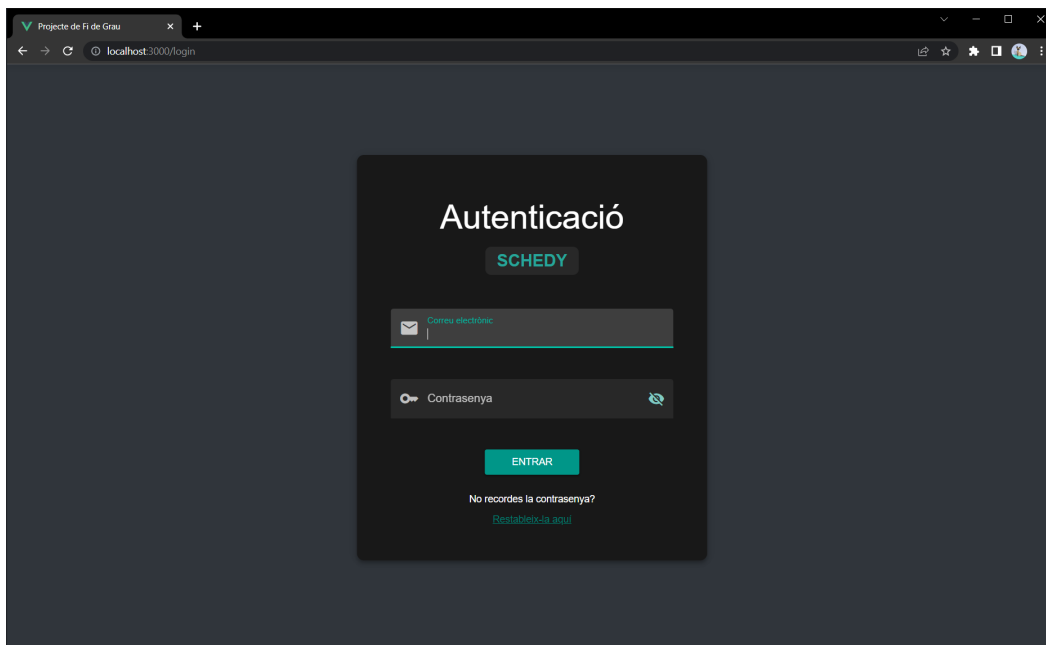


Figura 10.1: Interfície resultant d'autenticació.

A la figura 10.2 es pot veure la interfície resultant en la qual un usuari pot realitzar, principalment, les accions següents:

- Entrar l'adreça de correu electrònic del seu usuari per tal de restablir-ne la contrasenya.
- Prémer l'enllaç “Autentica't aquí” per tal de procedir a autenticar-se.

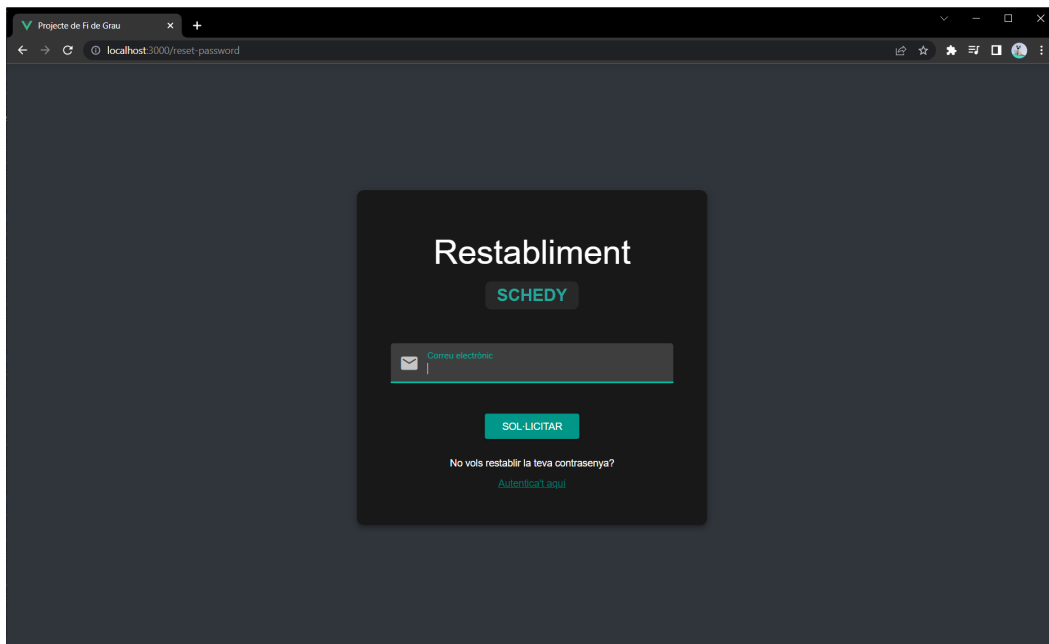


Figura 10.2: Interfície resultant de restabliment de contrasenya.

A la figura 10.3 es pot veure la interfície resultant en la qual un usuari pot realitzar, principalment, les accions següents:

- Informar-se de que se li ha enviat un correu electrònic a través del qual pot restablir la contrasenya del seu usuari.
- Prémer l'enllaç “Fes-ho aquí” per tal de procedir a autenticar-se.

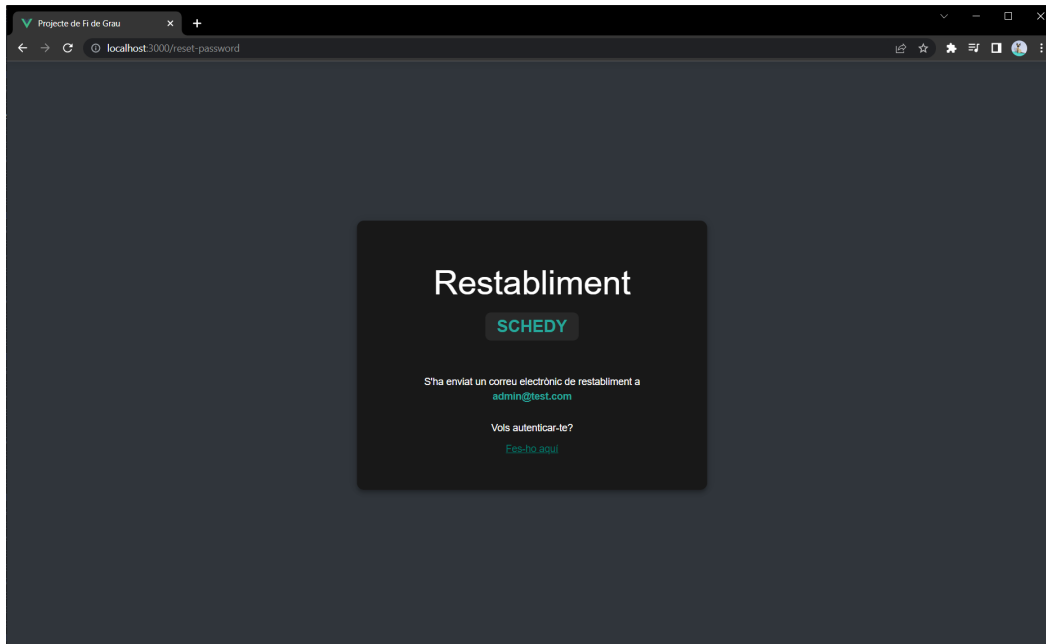


Figura 10.3: Interfície resultant de confirmació d'enviament d'un *email* per al restabliment de la contrasenya.

A la figura 10.4 es pot veure la interfície resultant en la qual un usuari pot realitzar, principalment, l'acció següent:

- Entrar dues vegades una contrasenya per tal d'escollir-ne una nova per al seu usuari.

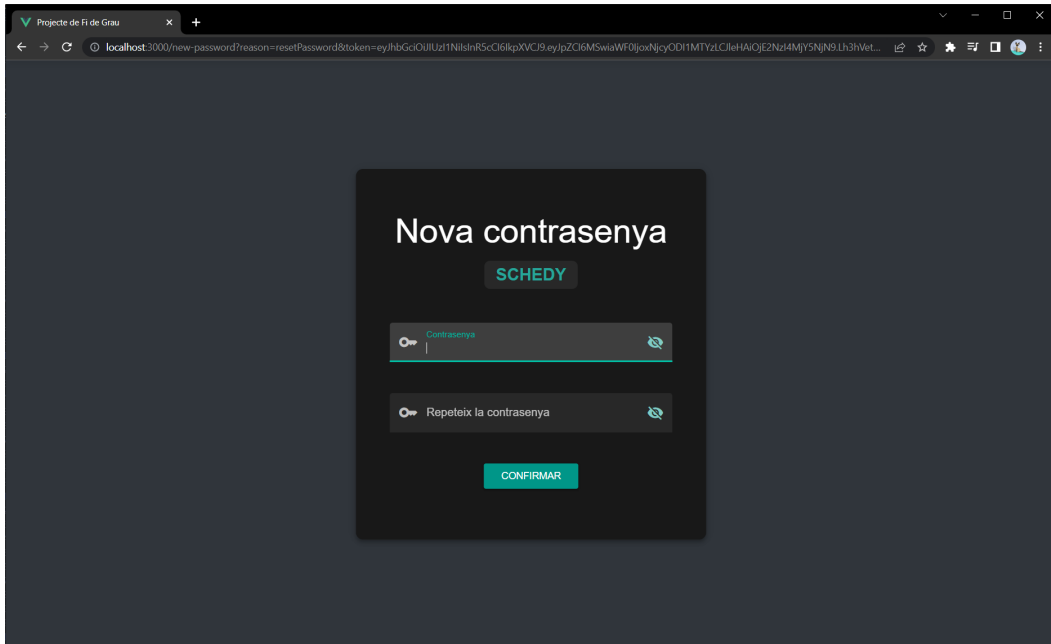


Figura 10.4: Interfície resultant d'elecció d'una nova contrasenya.

A la figura 10.5 es pot veure la interfície resultant en la qual un usuari pot realitzar, principalment, les acció següent:

- Prémer el botó “Sortir” situat a la part dreta de la barra superior per tal d’obrir un diàleg de confirmació a través del qual pot tancar la sessió.

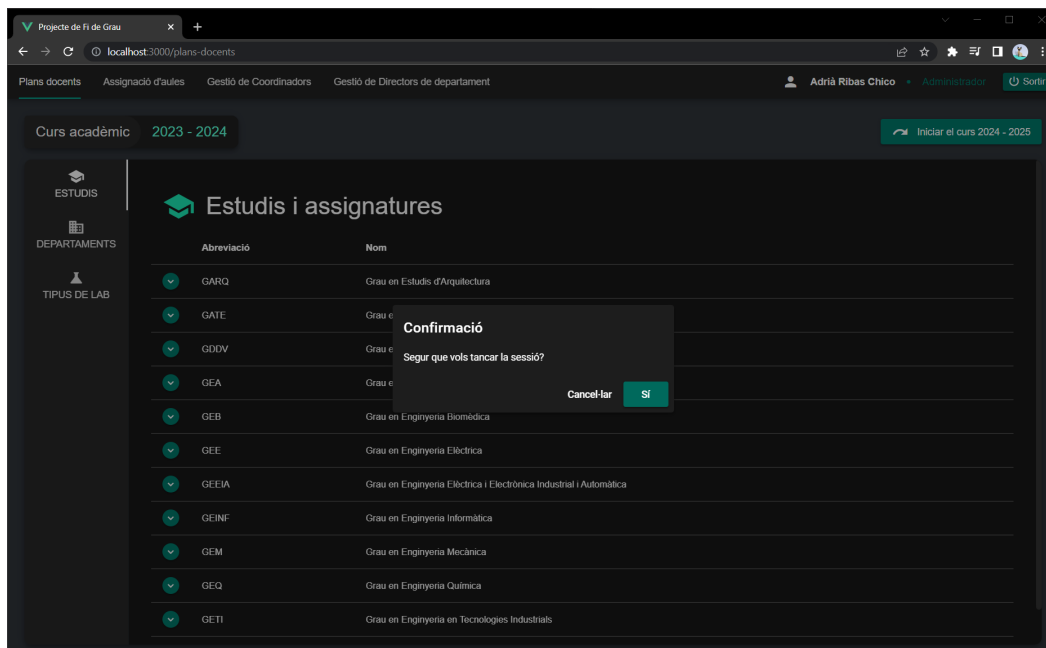


Figura 10.5: Interfície resultant de tancament de sessió.

10.2.3 Gestió del pla docent

En aquesta subsecció, s'exposaran els resultats obtinguts pel que fa a la gestió del pla docent.

A la figura 10.6 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Seleccionar el fitxer d'un pla docent del seu ordinador per tal de pujar-lo i inicialitzar, d'aquesta manera, el curs acadèmic en qüestió.

Cal destacar que es tracta del cas en què encara no s'ha pujat mai cap pla docent al sistema. En la situació d'estar pujant el pla docent del curs acadèmic següent, també es dóna la possibilitat de cancel·lar i tornar a la gestió del pla docent actual.

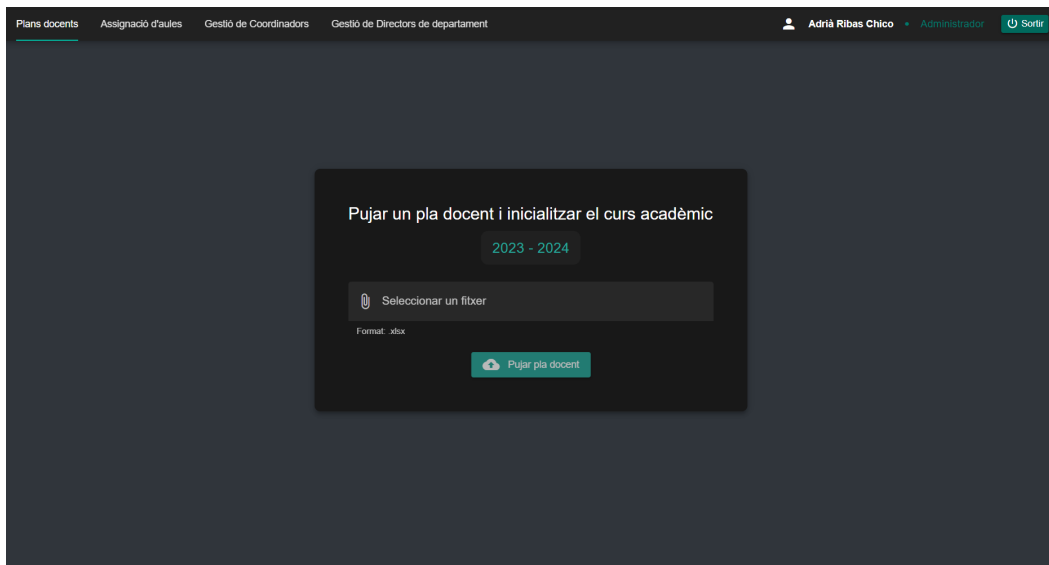


Figura 10.6: Interfície resultant de la pujada d'un pla docent.

A la figura 10.7 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Visualitzar els estudis del pla docent actual.
- Prémer un botó de desplegament per tal de visualitzar les assignatures impartides en un curs concret de l'estudi en qüestió.
- Alternar entre els cursos de l'estudi en qüestió per tal de visualitzar les assignatures corresponents.
- Situar el ratolí sobre l'abreviació d'una àrea o departament d'una assignatura per tal de visualitzar-ne el nom complet.
- Prémer el botó de modificació d'una assignatura per tal de procedir a fer-ho.

Cal destacar que, igual que passa en les figures 10.10 i 10.11, també es possibilita tant alternar entre la informació del pla docent com procedir a iniciar el curs acadèmic següent.

Codi	Nom	Quad.	Crèdits	Àrees	Tipus lab.	G. grans	G. mitjans	G. petits	
3105G00033	Instal·lacions 1	1	6	CONSTR → ARDECO	-	1	1	4	✎
3105G00037	Estructures 2	1	3	MMCTE → EMECCI	INFOR	1	1	3	✎
3105G11011	Organització, planificació i equips d'obra 1	1	6	CONSTR → ARDECO	AULA	1	1	1	✎
3105G11015	Amidaments i pressupostos 1	1	6	CONSTR → ARDECO	INFOR	1	1	1	✎
3105G11017	Valoracions i taxacions	1	3	CONSTR → ARDECO	-	1	1	0	✎
3105G11031	Construcció 5	1	6	CONSTR → ARDECO	AULA	1	1	1	✎
3105G00034	Instal·lacions 2	2	6	CONSTR → ARDECO	-	1	1	4	✎

Figura 10.7: Interfície resultant de visualització i gestió dels estudis i assignatures d'un pla docent.

A la figura 10.8 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Modificar el nombre de grups de cada tipus de l'assignatura en qüestió.
- Desassignar àrees concretes de l'assignatura en qüestió.
- Accedir a l'assignació d'una altra àrea.
- Situar el ratolí sobre l'abreviació d'un departament per tal de visualitzar-ne el nom complet.
- Assignar múltiples tipus de laboratori a l'assignatura en qüestió mitjançant el desplegable.
- Desassignar tots els tipus de laboratori de l'assignatura en qüestió.
- Desassignar un tipus de laboratori concret de l'assignatura en qüestió.
- Prémer el botó “Guardar” per tal de fer efectius els canvis.
- Prémer el botó “Cancel·lar” per tal de descartar els canvis.

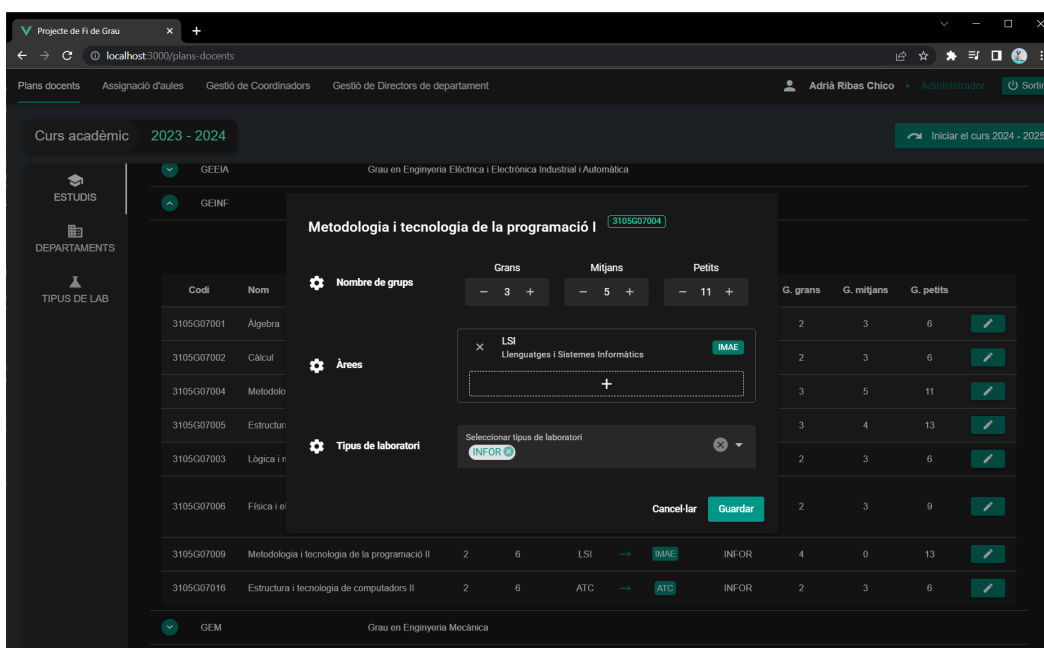


Figura 10.8: Interfície resultant de modificació d'una assignatura.

A la figura 10.9 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Obrir el desplegable “Departament” per tal d’escollir el departament que conté l’àrea que vol assignar a l’assignatura en qüestió.
- Obrir el desplegable “Àrea” per tal d’escollir l’àrea que vol assignar a l’assignatura en qüestió.

Cal destacar que el desplegable “Àrea” està bloquejat mentre no es selecciona un departament. A més a més, aquest desplegable no conté àrees que ja estan assignades a l’assignatura en qüestió i, en cas que només n’hi hagi una, es selecciona automàticament després d’haver escollit el departament. El botó “Afegir” també romandrà bloquejat mentre no s’hagi escollit una àrea concreta.

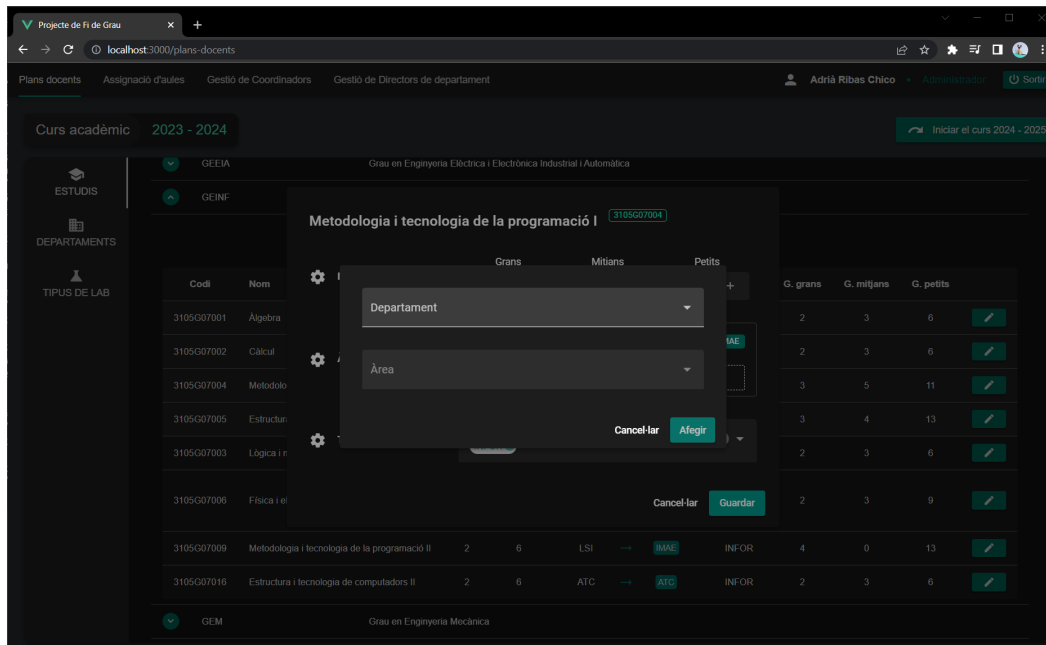


Figura 10.9: Interfície resultant de selecció d’una àrea per assignar-la a una assignatura.

A la figura 10.10 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Visualitzar els departaments del pla docent actual.
- Prémer un botó de desplegament per tal de visualitzar les àrees pertanyents al departament en qüestió.

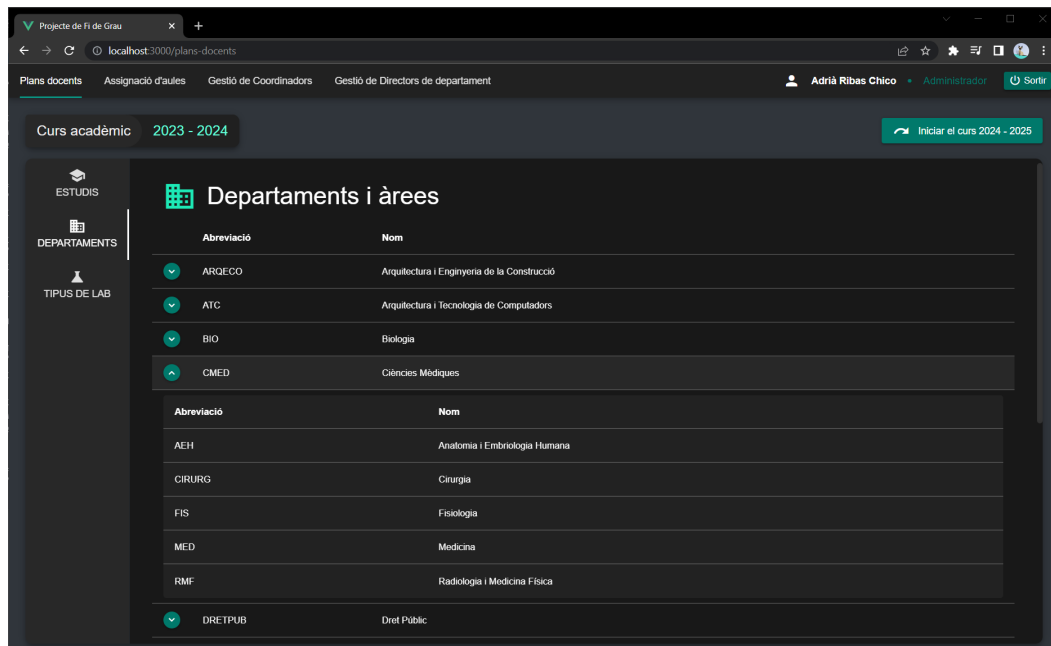
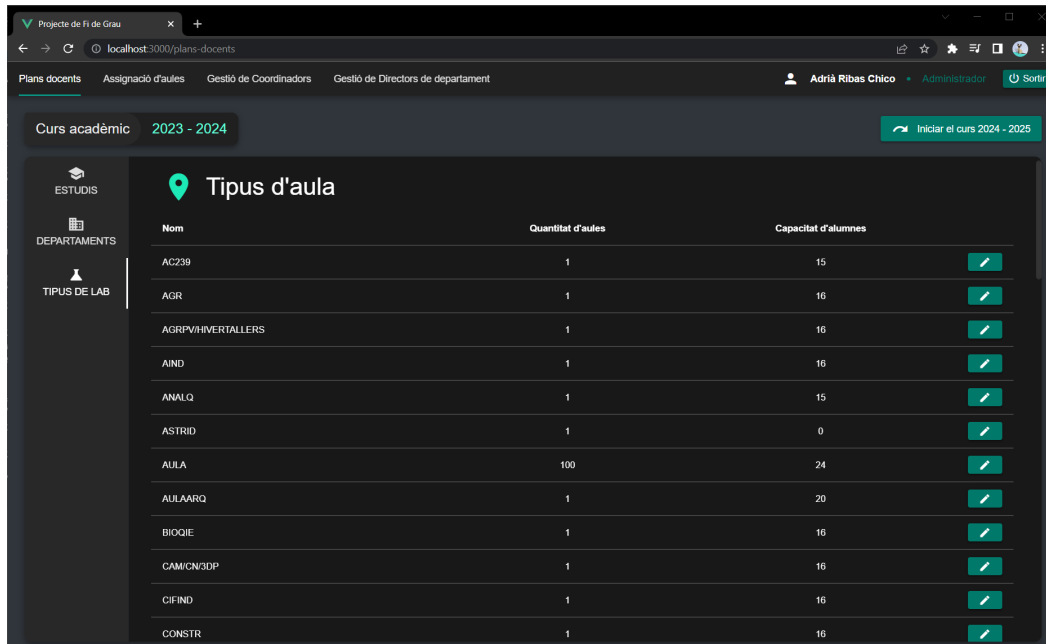


Figura 10.10: Interfície resultant de visualització dels departaments i àrees d'un pla docent.

A la figura 10.11 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Visualitzar els tipus de laboratori del pla docent actual.
- Prémer el botó de modificació d'una tipus de laboratori per tal de procedir a fer-ho.



The screenshot shows a web application interface for managing laboratory types. The interface is in a dark theme and includes a navigation menu on the left with options like 'ESTUDIS', 'DEPARTAMENTS', and 'TIPUS DE LAB'. The main content area is titled 'Tipus d'aula' and displays a table with the following data:

Nom	Quantitat d'aules	Capacitat d'alumnes	
AC239	1	15	
AGR	1	16	
AGRPV/IVERTALLERS	1	16	
AIND	1	16	
ANALQ	1	15	
ASTRID	1	0	
AULA	100	24	
AULAARQ	1	20	
BIOQIE	1	16	
CAM/CN/3DP	1	16	
CIFIND	1	16	
CONSTR	1	16	

Figura 10.11: Interfície resultant de visualització i gestió dels tipus de laboratori d'un pla docent.

A la figura 10.12 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Modificar la quantitat d'aules del tipus de laboratori en qüestió.
- Modificar la capacitat d'alumnes del tipus de laboratori en qüestió.
- Prémer el botó “Guardar” per tal de fer efectius els canvis.
- Prémer el botó “Cancel·lar” per tal de descartar els canvis.

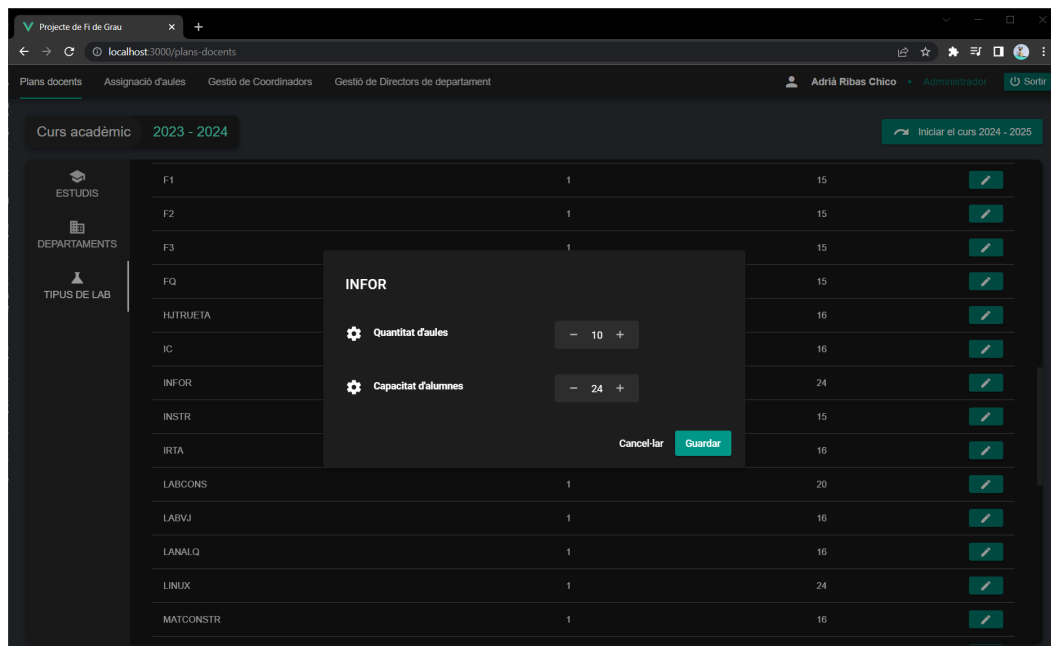


Figura 10.12: Interfície resultant de modificació d'un tipus de laboratori.

10.2.4 Gestió d'usuaris

En aquesta subsecció, s'exposaran els resultats obtinguts pel que fa a la gestió d'usuaris.

Aquests resultats corresponen a les interfícies de gestió de Coordinadors i la seva assignació a estudis. No obstant això, també són vàlids per a la gestió de Directors i la seva assignació a departaments, per a la gestió de Responsables de docència i la seva assignació a àrees i per a la gestió de Professors.

A la figura 10.13 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Seleccionar un usuari Coordinador mitjançant el desplegable “Seleccionar usuari” i prémer el botó d'assignació per tal d'assignar-lo a un dels estudi lliures.
- Prémer el botó de desassignació per tal de desassignar un usuari Coordinador d'un estudi.
- Prémer el botó “GESTIÓ DELS USUARIS” per tal d'accedir a la gestió dels usuaris Coordinadors.

Cal destacar que un usuari Coordinador que ja estigui assignat a un estudi no apareix als desplegable. A més a més, el botó d'assignació corresponent a un desplegable romandrà bloquejat mentre no hi hagi cap usuari seleccionat.

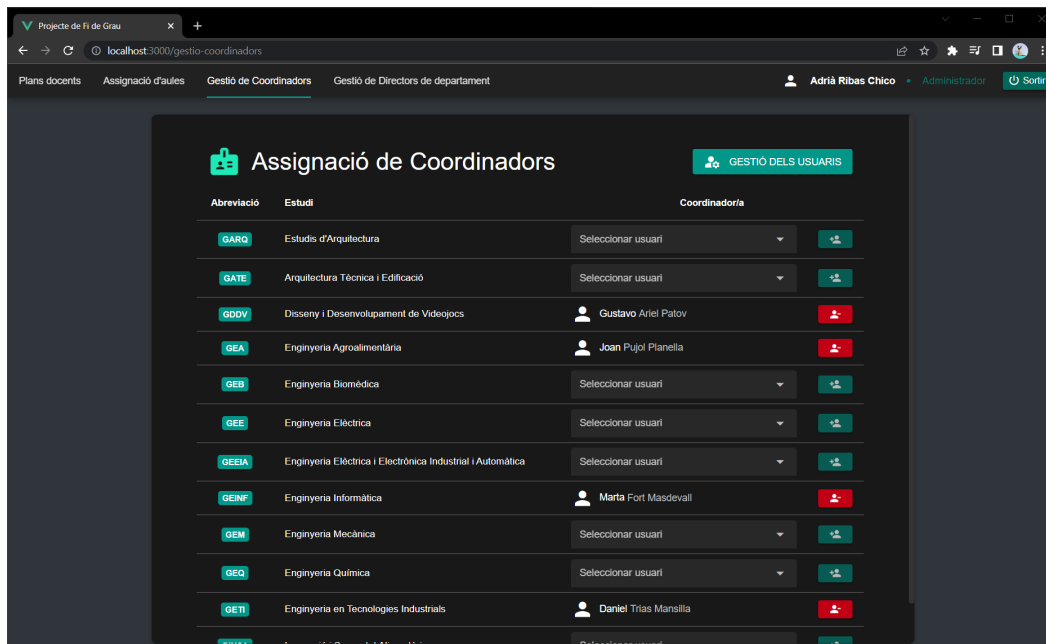


Figura 10.13: Interfície resultant d'assignació de Coordinadors a estudis.

A la figura 10.14 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Prémer el botó “NOU USUARI” per tal d’accedir a la creació d’un nou usuari Coordinador.
- Prémer el botó “Reenviar correu” per tal de tornar a enviar un correu electrònic d’activació a l’usuari en qüestió.
- Prémer el botó “Eliminar” per tal d’eliminar l’usuari en qüestió.

Cal destacar que només apareix l’opció de reenviar el correu electrònic d’activació als usuaris que encara no han estat activats. A més a més, després de prémer el botó que els envia, queda bloquejat uns segons per tal d’evitar l’enviament massiu de correus. També s’indica si els usuaris estan o no assignats a un estudi i, en cas afirmatiu, a quin.

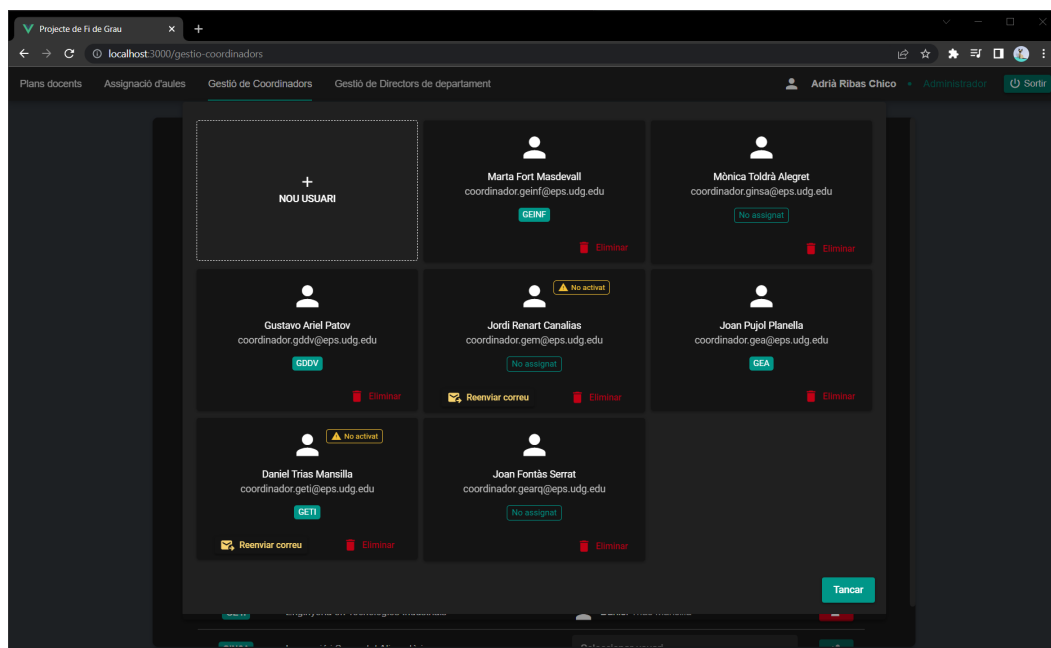


Figura 10.14: Interfície resultant de la gestió de Coordinadors.

A la figura 10.15 es pot veure la interfície resultant en la qual un Administrador pot realitzar, principalment, les accions següents:

- Entrar el nom, cognoms i adreça de correu electrònic de l'usuari Coordinador que s'estigui creant.
- Prémer el botó “Crear” o la tecla “enter” per tal de crear l'usuari corresponent.
- Prémer el botó “Cancel·lar” per tal d'abortar el procés de creació.

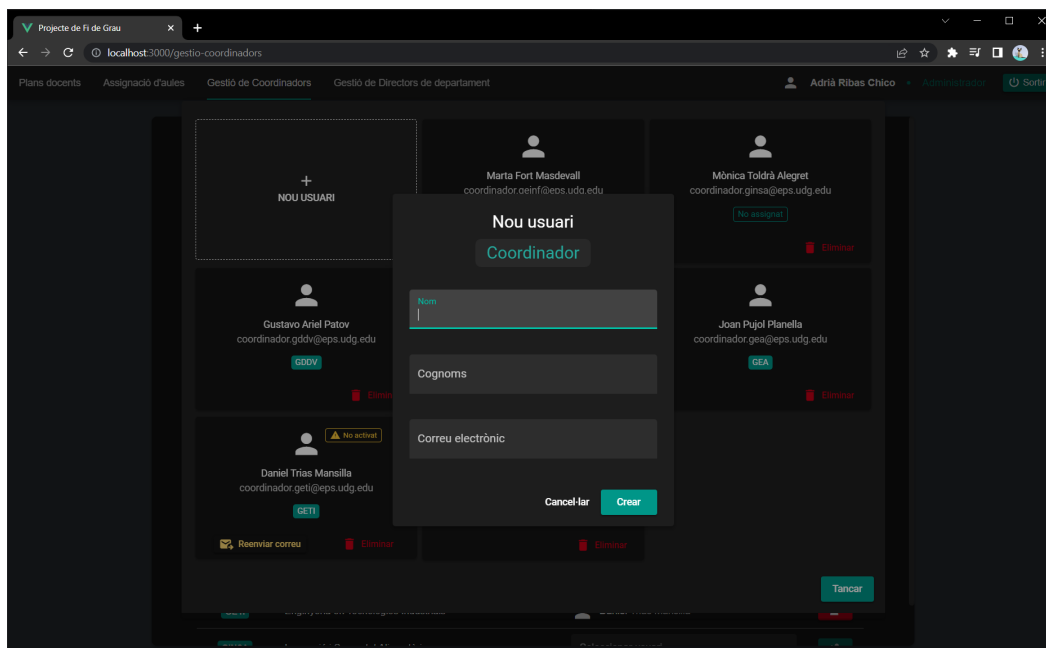


Figura 10.15: Interfície resultant de creació d'un nou Coordinador.

10.2.5 Consulta i elaboració d'horaris

En aquesta subsecció, s'exposaran els resultats obtinguts pel que fa a la consulta i elaboració d'horaris.

A la figura 10.16 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Prémer el botó de visualització per tal d'accedir a la visualització de l'horari en qüestió del grau que gestiona o bé d'un dels estudis amb els quals comparteix assignatures.
- Prémer el botó de modificació per tal d'accedir a la modificació de l'horari en qüestió del grau que gestiona.

Aquest resultat també és vàlid per a l'accés a la visualització dels horaris dels estudis corresponents per part d'un Director de departament o d'un Professor.

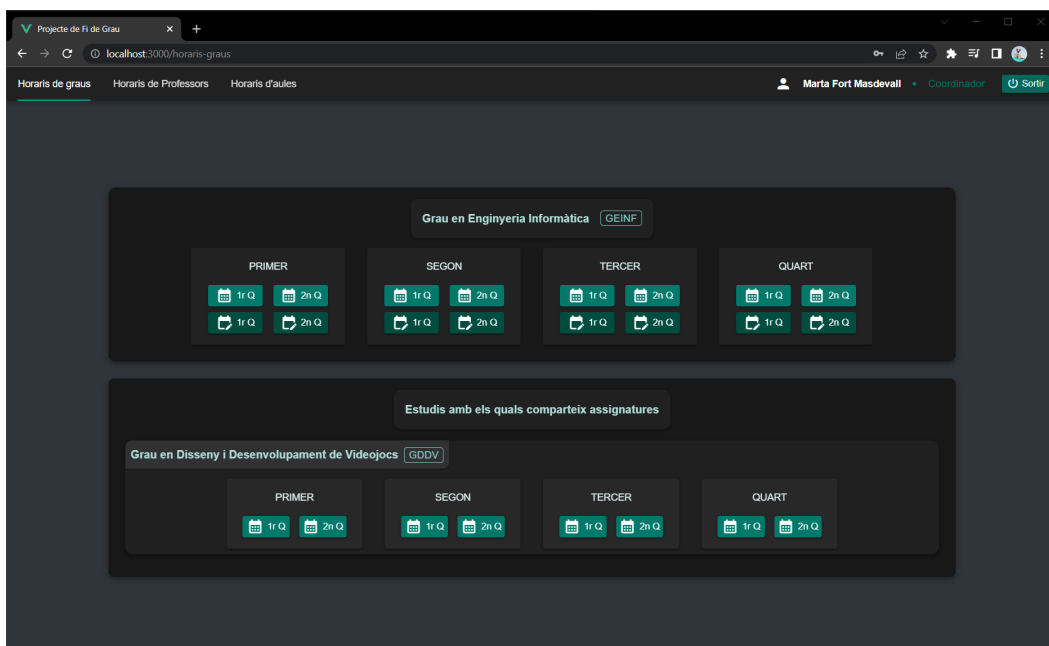


Figura 10.16: Interfície resultant d'accés a visualitzar o a modificar un horari d'un estudi.

A la figura 10.17 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Informar-se de quin horari està visualitzant mitjançant els *breadcrumbs* de la part superior esquerra.
- Prémer l'icona de la part esquerra dels *breadcrumbs* per tal de tornar a l'accés a horaris.
- Visualitzar la distribució de l'horari en una vista setmanal concreta.
- Alternar entre vistes setmanals mitjançant el grup de botons de la part superior dreta.
- Informar-se sobre el color que representa cada tipus de grup mitjançant la llegenda de la part superior central.
- Situar el ratolí sobre un bloc horari o per tal d'informar-se'n de l'hora d'inici, l'hora de fi i de la duració.
- Prémer un bloc horari per tal d'accedir a la visualització de tota la seva informació.

Cal destacar que els blocs d'horaris dels grups pertanyents a assignatures compartides només són visibles si el grup en qüestió està assignat, com a mínim, a l'estudi del qual és l'horari. A més a més, totes les possibilitats que involucren blocs horaris, també serveixen per a blocs horaris genèrics.

Aquest resultat també és vàlid per a la visualització d'horaris tant de Professors com d'aules.

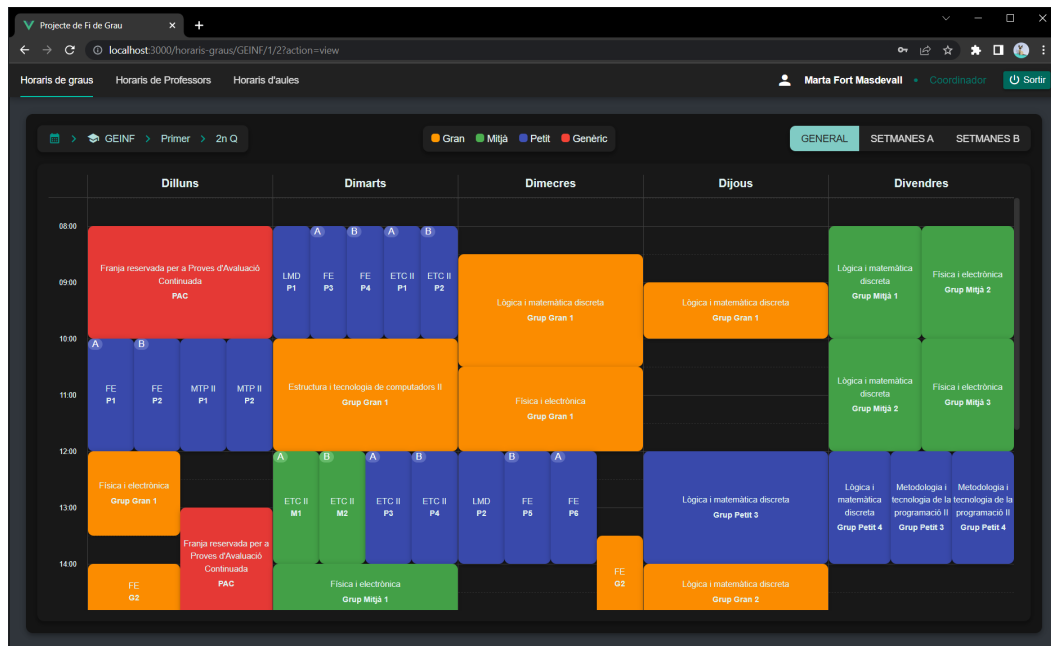


Figura 10.17: Interfície resultant de visualització d'un horari d'un estudi.

A la figura 10.18 es pot veure com queda la distribució del mateix horari de la figura anterior si l'usuari Coordinador a canviat a la vista de setmanes B.

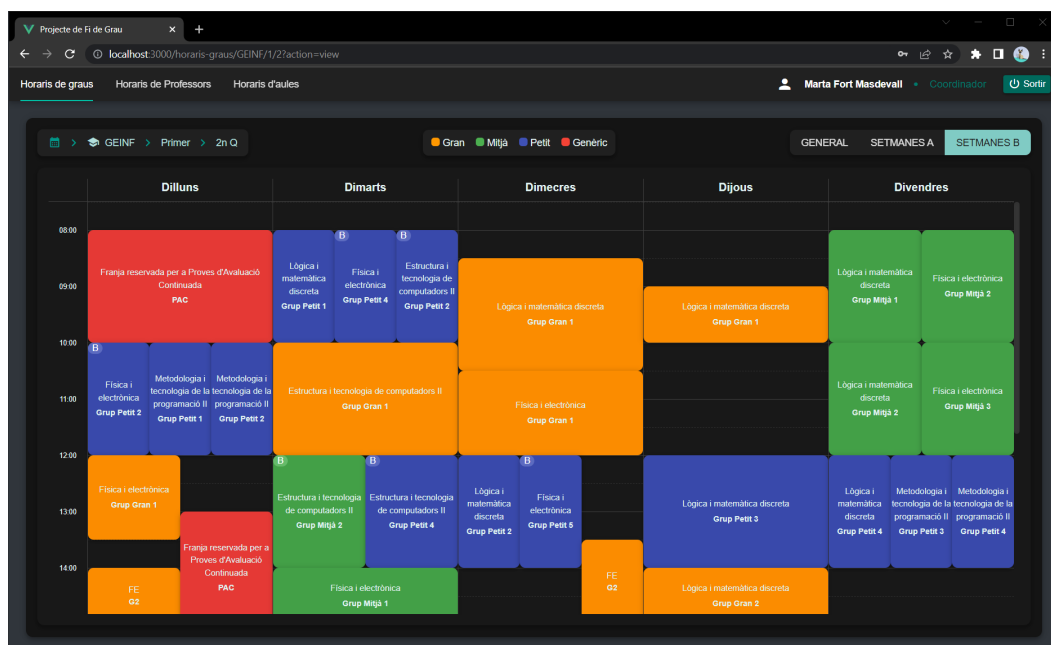


Figura 10.18: Interfície resultant de la visualització d'un horari horari d'un estudi en la vista de setmanes B.

A la figura 10.19 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Visualitzar el tipus i el número del grup al qual pertany el bloc horari.
- Visualitzar el nom de l'assignatura a la qual pertany el grup del bloc horari.
- Visualitzar la informació temporal del bloc horari: dia, hora d'inici, hora de fi, duració i setmana.
- Visualitzar el nom de l'espai assignat al bloc horari.
- Visualitzar el nom i els cognoms del Professor assignat al bloc horari.
- Visualitzar les àrees i departaments als quals està assignada l'assignatura a la qual pertany el grup del bloc horari.
- Visualitzar l'abreviació i el nom dels estudis als quals està assignat el grup del bloc horari.
- Visualitzar els tipus de laboratori assignats a l'assignatura a la qual pertany el grup del bloc horari.

Cal destacar que els estudis als quals està assignat el grup del bloc horari només apareixen en cas que l'assignatura a la qual pertany el grup sigui compartida. De la mateixa manera, els tipus de laboratori assignats a l'assignatura en qüestió només apareixen en cas que es tracti d'un grup petit.

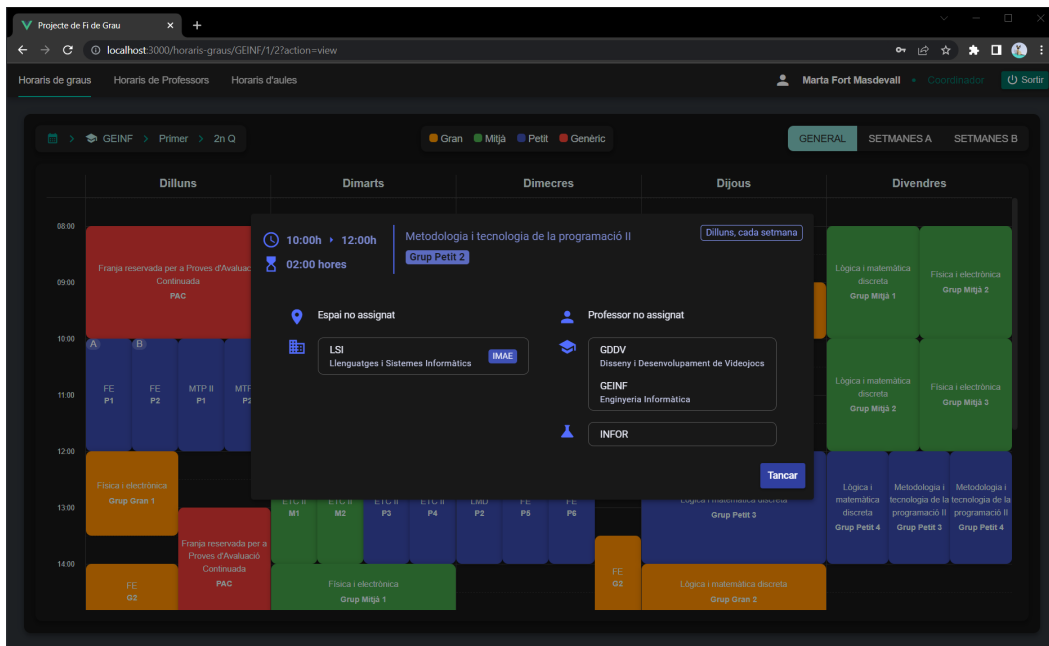


Figura 10.19: Interfície resultant de visualització de la informació d'un bloc horari.

A la figura 10.20 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Visualitzar l'etiqueta del bloc horari genèric.
- Visualitzar la subetiqueta del bloc horari genèric.
- Visualitzar la informació temporal del bloc horari genèric: dia, hora d'inici, hora de fi, duració i setmana.

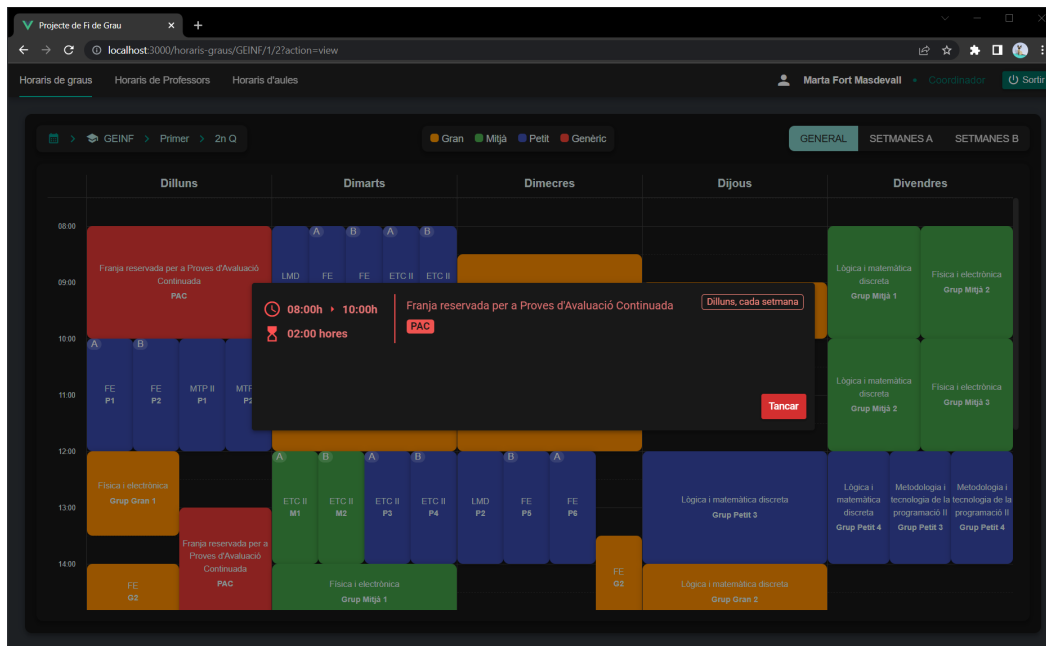


Figura 10.20: Interfície resultant de visualització de la informació d'un bloc horari genèric.

A la figura 10.21 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Arrossegar l'icona de la intersecció entre els dos panells per tal de canviar-ne les proporcions.
- Plegar o desplegar les opcions disponibles al panell de la dreta.
- Informar-se dels blocs horaris pendents de col·locar respecte del total de blocs d'una assignatura o bé genèrics.
- Arrossegar un bloc horari no col·locat i deixar-lo anar a l'horari per tal d'afegir-lo.
- Arrossegar un bloc horari col·locat i deixar-lo anar a la zona dels "no col·locats" per tal de treure'l.
- Arrossegar un bloc horari col·locat i deixar-lo anar a l'horari per tal de moure'l.
- Agafar amb el ratolí un dels extrems verticals d'un bloc horari col·locat i arrossegar el ratolí per tal d'allargar-lo o escurçar-lo, de manera que es modifiqui la informació temporal corresponent.
- Prémer un bloc horari (col·locat o no col·locat) per tal d'accedir a la modificació de tota la seva informació.

Cal destacar que les opcions de visualització vistes a la figura 10.17 també estan disponibles en aquesta interfície. A més a més, totes les possibilitats que involucren blocs horaris, també serveixen per a blocs horaris genèrics.

The screenshot displays a web-based interface for managing a study schedule. The main area is a grid with days of the week (Dilluns, Dimarts, Dimecres, Dijous, Divendres) on the x-axis and time slots (08.00, 09.00, 10.00, 11.00, 12.00, 13.00, 14.00) on the y-axis. A red block labeled 'PAC' is positioned on Monday from 09.00 to 11.00. An orange block labeled 'Àlgebra Grup Gran 1' is on Tuesday from 10.00 to 12.00. A blue block labeled 'Àlgebra Grup Petit 3' is on Thursday from 12.00 to 14.00. On Wednesday, there are two blue blocks labeled 'MTP P1' and 'MTP P3' from 09.00 to 10.00, and a green block labeled 'ETC M1' from 10.00 to 11.00. The right sidebar contains a 'Configuració' panel with 'Blocs horaris' and 'Càlcul' sections. The 'Blocs horaris' section lists 'Genèries' (1 de 2 blocs col·locats) with a red 'PAC' block (02.00h) and 'Àlgebra' (2 de 13 blocs col·locats) with various colored blocks: orange (GG 1, GG 2), green (GM 1, GM 2, GM 3), blue (GP 1, GP 2, GP 4, GP 5), and blue (GP 6). The 'Càlcul' section shows 0 de 11 blocs col·locats with orange (GG 1, GG 2) and green (GM 1, GM 2) blocks.

Figura 10.21: Interfície resultant de modificació d'un horari d'un estudi.

A la figura 10.22 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Modificar l'hora d'inici i l'hora de fi del bloc horari mitjançant l'entrada manual o bé l'ús d'un rellotge desplegable a través del botó de la part dreta d'aquests camps.
- Modificar la duració del bloc horari.
- Canviar el dia en què el bloc horari està situat o esborrar-lo per tal de treure'l de l'horari.
- Canviar la setmana en què el bloc horari està assignat.
- Dessassignar un estudi del grup del bloc horari.
- Assignar un estudi al grup del bloc horari mitjançant el desplegable "Afegir estudi".
- Prémer el botó "Guardar" per tal de fer efectius els canvis.
- Prémer el botó "Cancel·lar" per tal de descartar els canvis.

Cal destacar que els camps tant de les hores com el de la duració s'assisteixen i actualitzen automàticament i no deixen entrar valors que es surtin dels rangs permesos. La duració es prioritza sobre les hores. A més a més, al desplegable dels estudis només hi apareixeran aquells amb què l'assignatura en qüestió estigui compartida.

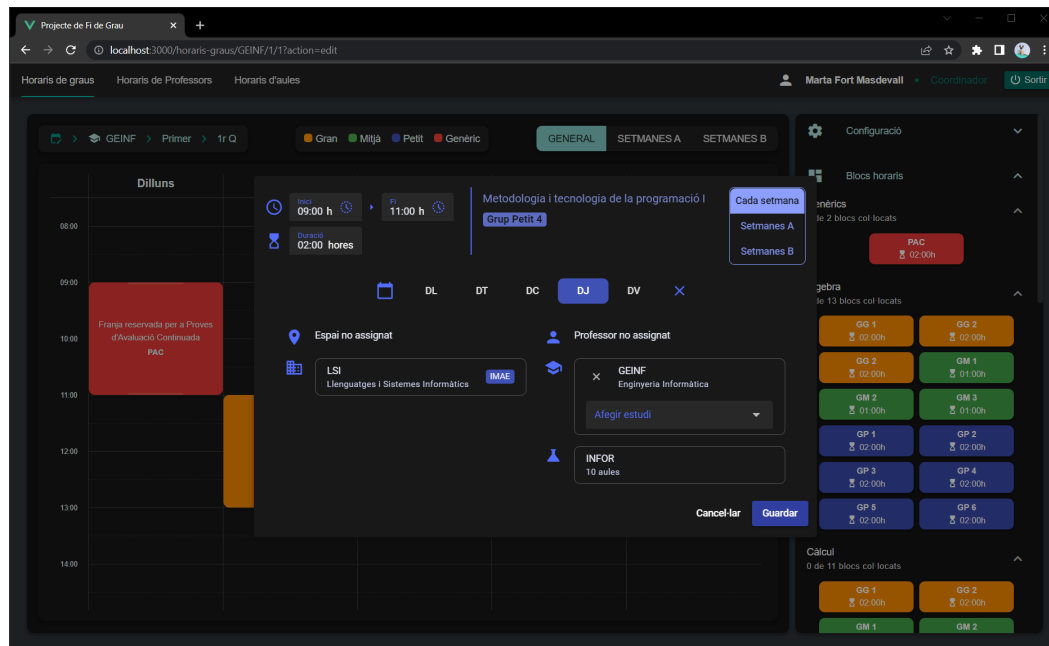


Figura 10.22: Interfície resultant de modificació d'un bloc horari.

A la figura 10.23 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Modificar la informació temporal del bloc horari genèric: dia, hora d'inici, hora de fi, duració i setmana.
- Modificar l'etiqueta i la subetiqueta del bloc horari genèric.
- Prémer el botó “Guardar” per tal de fer efectius els canvis.
- Prémer el botó “Cancel·lar” per tal de descartar els canvis.

Cal destacar que les opcions de modificació de la informació temporal coincideixen amb les que s'han vist a la figura 10.22. A més a més, el camp de la subetiqueta no permet l'entrada de més de 10 caràcters.

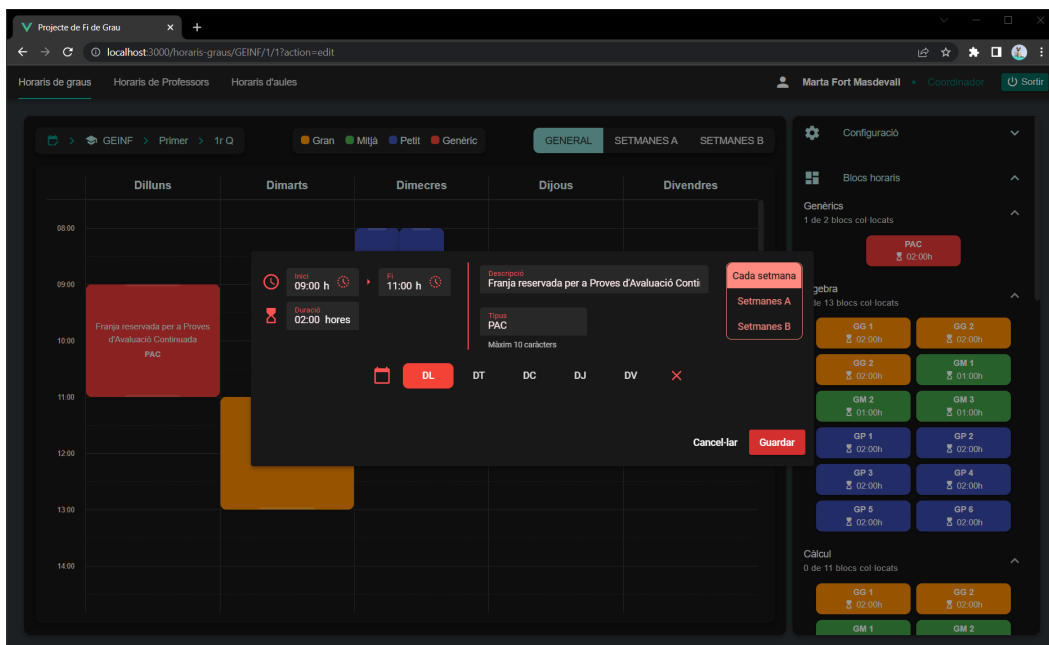


Figura 10.23: Interfície resultant de modificació d'un bloc horari genèric.

A la figura 10.24 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, l'acció següent:

- Situar el ratolí sobre un tipus de laboratori marcat com a solapat amb una icona vermella per tal de visualitzar els cursos de cadascun dels estudis implicats en el solapament.

Cal destacar que també es mostra un apartat anomenada “NO ASSIGNATS” per tal de comptabilitzar els blocs horaris d'assignatures compartides el grup dels quals no està assignat a cap estudi.

Aquest format també serveix per marcar els altres elements amb risc de solapament.

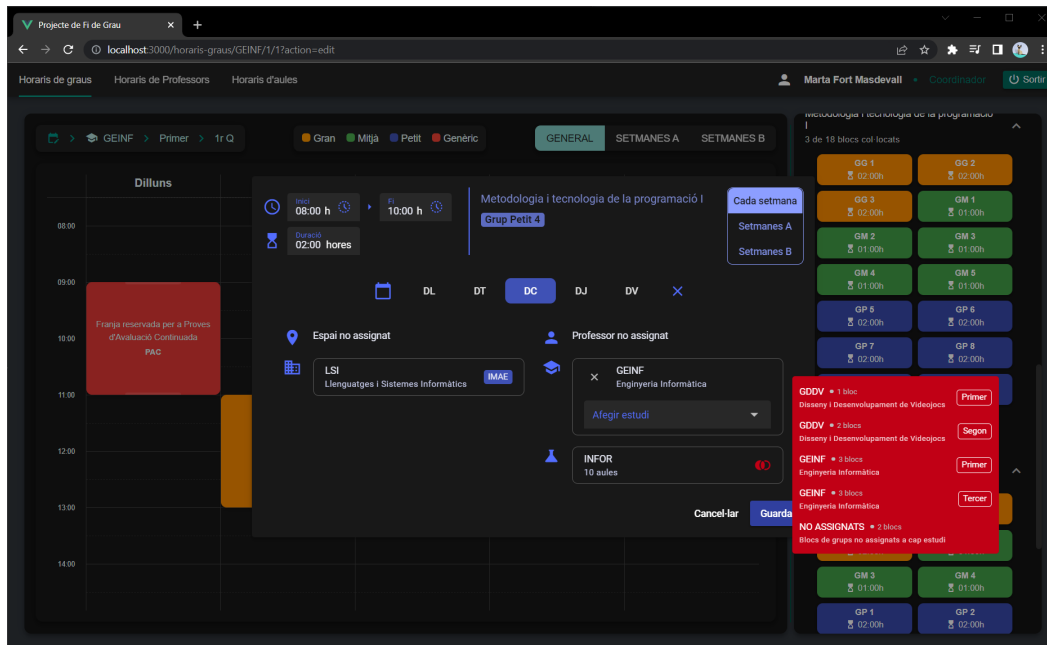


Figura 10.24: Interfície resultant de visualització de la informació detallada sobre els solapaments de tipus de laboratori.

A la figura 10.25 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Visualitzar els les franges horàries en què es produïria algun tipus de solapament si hi deixés anar el bloc horari que estigui arrossegant en un moment determinat.

Cal destacar que això també funciona mentre s'allarga o s'escurça un bloc horari.

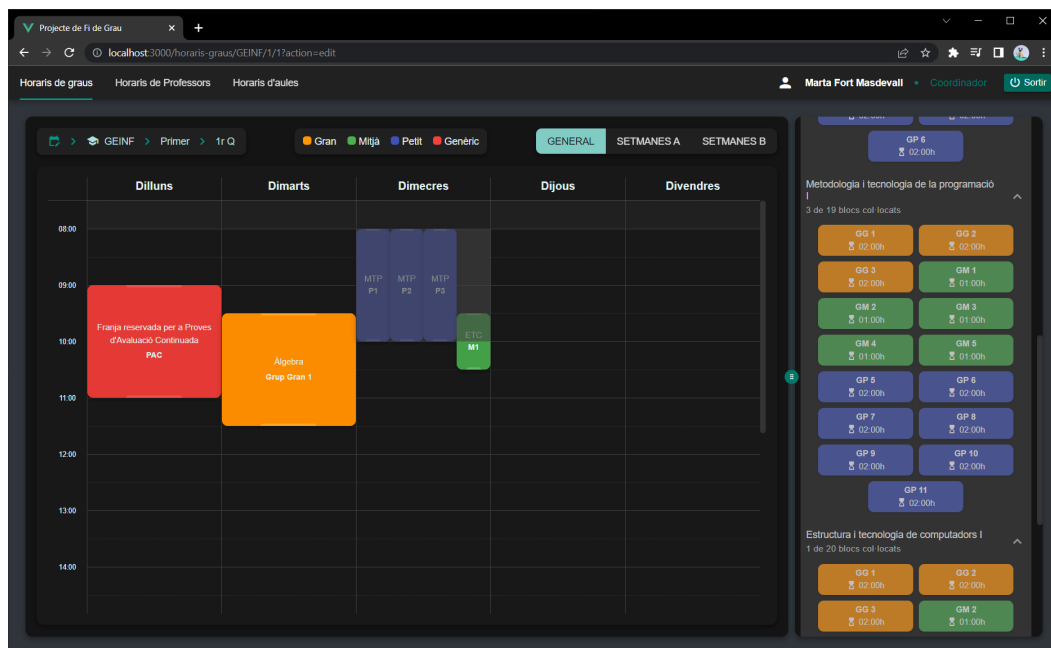


Figura 10.25: Interfície resultant de visualització de possibles solapaments a l'horari.

A la figura 10.26 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, l'acció:

- Visualitzar una marca vermella en els blocs horaris que provoquen solapaments.

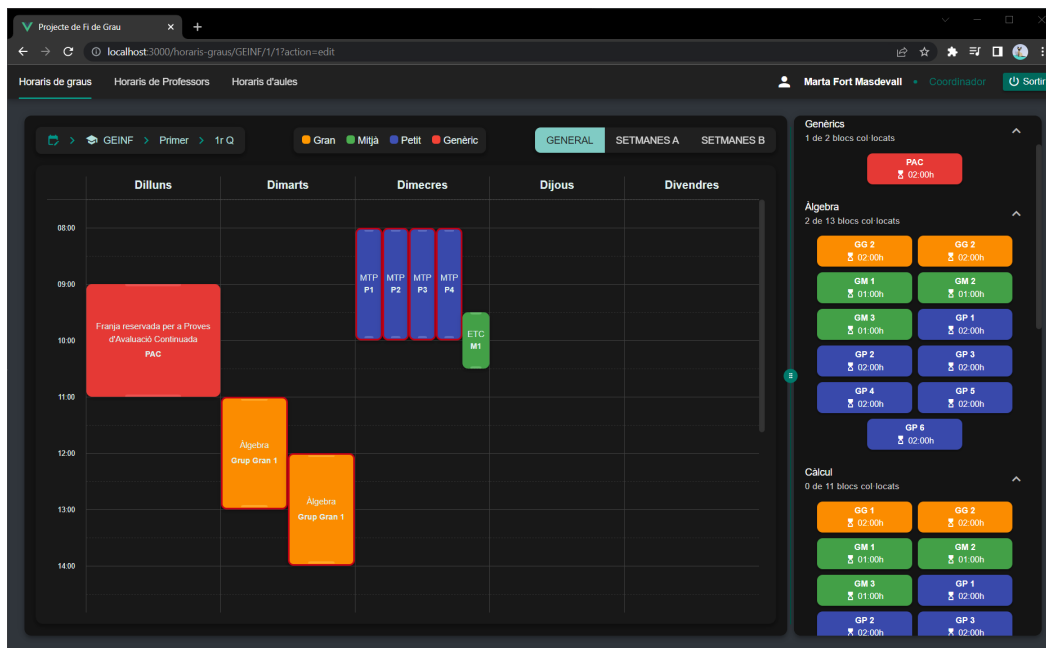


Figura 10.26: Interfície resultant de visualització de marques en els blocs horaris que es solapen.

A la figura 10.27 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Habilitar o deshabilitar el filtratge de blocs horaris d'assignatures compartides el grup dels quals no està assignat a cap estudi.
- Habilitar o deshabilitar el filtratge de blocs horaris d'assignatures compartides el grup dels quals està assignat només a altres estudis.
- Accedir a la gestió de blocs horaris.
- Accedir a la gestió de blocs horaris genèrics.
- Activar o desactivar la visualització de solapaments de blocs horaris sobre l'horari.
- Activar o desactivar la visualització de solapaments de tipus de laboratori sobre l'horari.
- Activar o desactivar la visualització de solapaments de Professors sobre l'horari.
- Activar o desactivar la visualització de solapaments d'aules sobre l'horari.

Cal destacar que aquesta figura està enfocada a presentar les funcionalitats de l'opció de configuració del panell de la dreta.

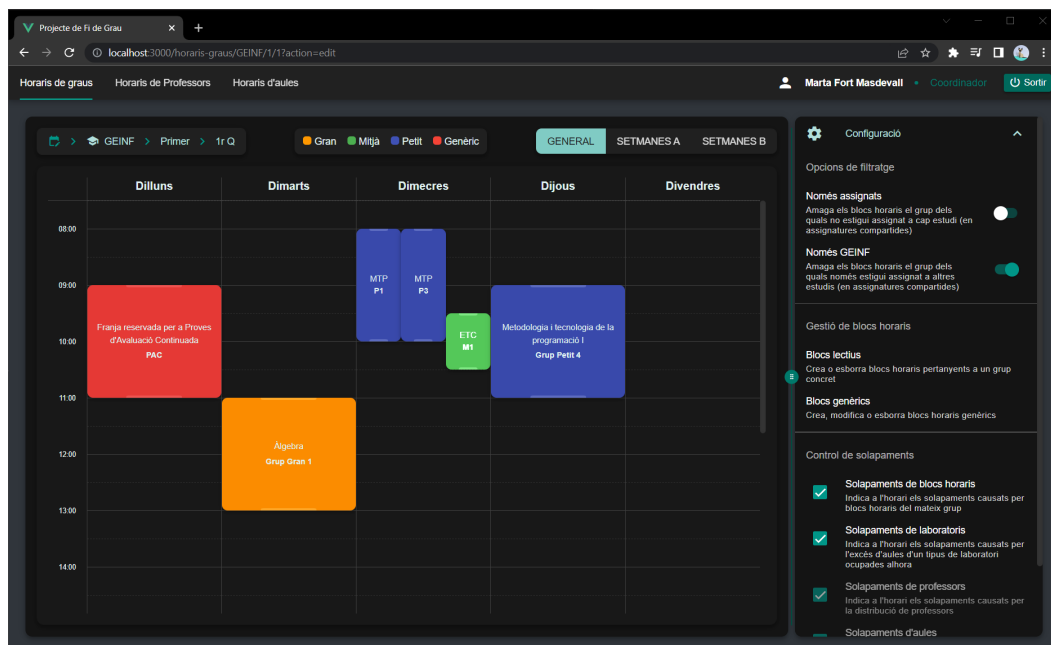


Figura 10.27: Interfície resultant de configuració de la modificació d'un horari.

A la figura 10.28 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Utilitzar el desplegable per tal de seleccionar l'assignatura de la qual es volen gestionar els blocs horaris.
- Alternar entre el tipus de grup dels quals es volen gestionar els blocs horaris.
- Eliminar un bloc horari d'un grup.
- Prémer el botó “+” per tal de crear un bloc horari per a un grup.
- Prémer el botó “Guardar” per tal de fer efectius els canvis.
- Prémer el botó “Cancel·lar” per tal de descartar els canvis.

Cal destacar que es pot alternar tant d'assignatura com de tipus de grup sense perdre els canvis per tal de fer tota la gestió desitjada i guardar-ne els canvis de manera conjunta.

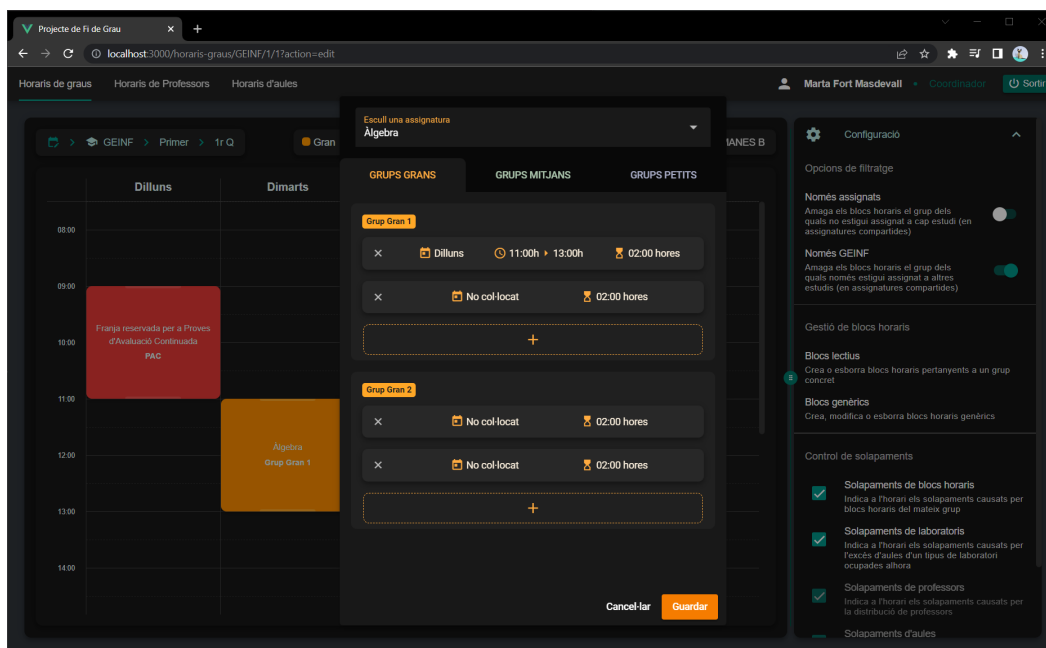


Figura 10.28: Interfície resultant de gestió de blocs horaris.

A la figura 10.29 es pot veure la interfície resultant en la qual un Coordinador pot realitzar, principalment, les accions següents:

- Eliminar un bloc horari genèric.
- Accedir a la modificació bàsica (etiqueta, subetiqueta i, en cas que no estigui col·locat, duració) d'un bloc horari genèric, amb la possibilitat de guardar o descartar els canvis temporals.
- Prémer el botó “+” per tal d'accedir a la creació d'un nou bloc horari genèric (molt semblant a la modificació).
- Prémer el botó “Guardar” per tal de fer efectius els canvis.
- Prémer el botó “Cancel·lar” per tal de descartar els canvis.

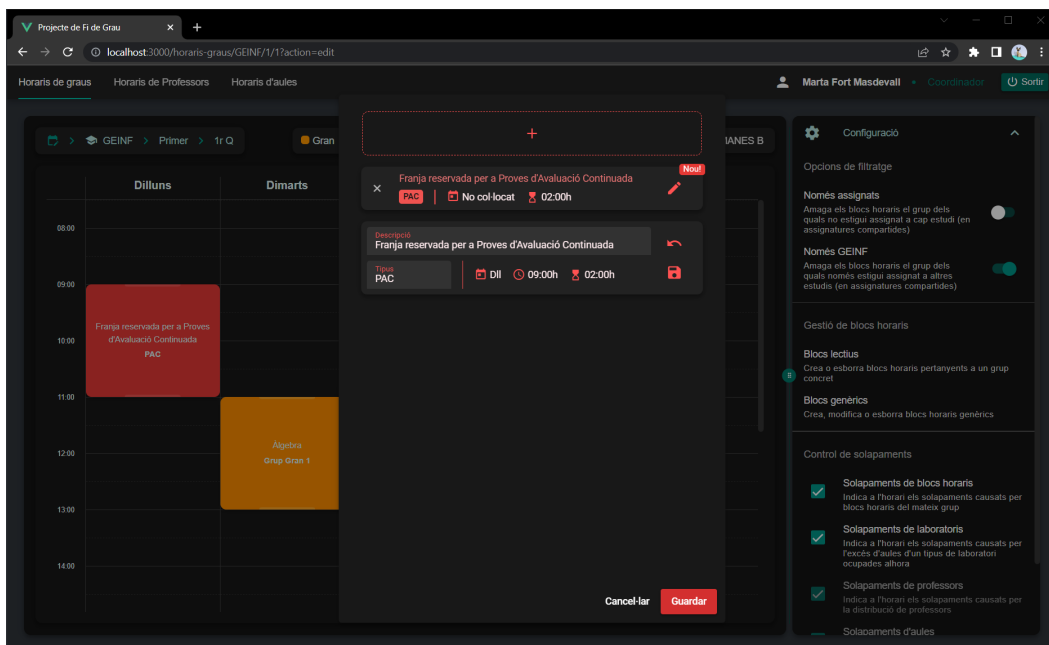


Figura 10.29: Interfície resultant de la gestió de blocs horaris genèrics.

Conclusions

En aquest capítol, es presentarà el grau d'assoliment dels objectius i requisits del projecte (veure secció 11.1) i se'n realitzarà una valoració personal (veure secció 11.2).

11.1 Assoliment dels objectius i els requisits

En aquesta secció, es farà una anàlisi comparativa entre els objectius i requisits plantejats inicialment respecte dels resultats obtinguts.

L'objectiu principal del projecte era desenvolupar una aplicació web que permetés l'elaboració dels horaris de l'EPS a través d'interfícies d'usuari còmodes, atractives i eficients. A més a més, havia de ser capaç de detectar els possibles solapaments que s'hi poguessin produir i avisar-ne l'usuari.

Abans d'arribar a aquest punt, però, havien d'assolir-se altres objectius essencials: obtenir la informació bàsica per al funcionament de l'aplicació a partir d'arxius de plans docents i oferir un control d'usuaris estructurat en un conjunt de rols.

Tal com s'ha vist en el capítol 10, els resultats obtinguts no només satisfan els objectius principals, sinó que també aporten alguns detalls i funcionalitats addicionals.

En general, es pot dir que s'ha aconseguit la meta del projecte. No obstant això, en el capítol 1 també s'indicaven altres objectius amb un pes inferior als anteriors. Més concretament, es tracta de la distribució de docència i de l'assignació d'aules i la visualització dels seus horaris. Tal com s'havia calculat en el capítol 6, si no s'acotaven aquests objectius, la finalització del projecte a temps no era possible. No obstant això, també s'havia esmentat que era important implementar l'aplicació pensant en fer una base de codi de qualitat, la qual en facilités un futur desenvolupament.

El resultat d'aquest últim punt també ha estat satisfactori, ja que la implementació de les interfícies dels horaris està basada en jerarquies de components altament reutilitzables. A més a més, les bases de la part corresponent a l'aplicació

servidor també estan pensades en aquest sentit.

Pel que fa als requisits, cal indicar que s'han aconseguit tots els que tenien màxima prioritat (veure capítol 5). A més, s'han assentat les bases per al futur desenvolupament dels requisits amb prioritats inferiors.

11.2 Valoració personal

En aquesta secció, es valorarà, en primera persona, l'experiència personal viscuda durant el transcurs del projecte.

Des que es va plantejar, de seguida vam veure que el projecte era molt ambiciós i que el més recomanable era desenvolupar-lo entre dues persones. Així i tot, vaig decidir enfrontar-m'hi sol per tal de no perdre'm cap detall sobre tot el que comportava, com ara la seva gestió, decisions d'arquitectura, la implementació d'una aplicació *back-end* i, sobretot, el desenvolupament del client d'una plataforma web, món al qual em vull dedicar i del qual encara no tenia cap mena d'experiència.

En aquest sentit, valoro molt positivament el procés de desenvolupament del projecte, ja que, gràcies als coneixements assolits durant la carrera, he pogut adquirir-ne de nous i aprendre sobre tecnologies que desconeixia.

Pel que fa als resultats obtinguts, n'estic molt orgullós i la realitat és que no m'esperava que fossin, des del meu punt de vista, tan satisfactoris. Igualment, crec que hi ha coses que es poden millorar i que ara faria de manera diferent.

Finalment, m'agradaria afegir que el projecte també m'ha ofert una perspectiva bastant "realista" sobre el que implica el desenvolupament d'un producte per a un client, en aquest cas la Marta.

Treball futur

En aquest capítol, es presentaran les idees de treball futur pensades pel projecte. Més concretament, se'n detallaran de bàsiques o essencials (veure secció 12.1) i d'addicionals (veure secció 12.2). En els dos casos, els punts s'ordenen de més prioritari a menys prioritari.

12.1 Ampliacions i millores bàsiques

En aquesta secció, es recolliran els aspectes bàsics que calen ampliar per tal de poder dur a terme una gestió d'horaris completa.

- Donar d'alta de Professors i assignar-los a blocs horaris per part dels Responsables de docència.
- Visualització dels horaris dels Professors per part d'ells mateixos, dels Coordinadors, dels Directors de departament i dels Responsables de docència.
- Control del solapament de Professors durant el procés de modificació dels horaris dels estudis per part dels Coordinadors.
- Visualització dels horaris dels estudis per part dels Directors de departament i dels Professors.
- Visualització dels horaris de les assignatures per part dels Professors.
- Assignació d'aules a blocs horaris per part dels Administradors, amb control de solapaments.
- Visualització dels horaris de les aules per part dels Coordinadors.
- Control del solapament d'aules durant el procés de modificació dels horaris dels estudis per part dels Coordinadors.

12.2 Ampliacions i millores addicionals

En aquesta secció, es contemplaran possibles ampliacions i millores addicionals que poden efectuar-se al projecte.

- Llançament de l'aplicació a producció.
- Sistema d'avisos i notificacions que permetés als usuaris assabentar-se dels esdeveniments importants relacionats amb ells. Algun exemple seria el sorgiment d'una incompatibilitat, l'assignació d'un professor o canvis en el pla docent, entre altres.
- Cerca d'usuaris per tal de comunicar-s'hi, ja sigui de manera interna o a través de correus electrònics. Això permetria facilitar les tasques i solucionar problemes còmodament.
- Possibilitar als Professors indicar les seves preferències horàries per tal que tenir-les en compte.
- Recuperació de la informació dels plans docents directament de les bases de dades de la universitat, en comptes de fer-ho a partir d'un fitxer.
- Realització de propostes d'horaris generats automàticament, tenint en compte les preferències dels Professors i evitant qualsevol tipus de solapament, entre altres aspectes.
- Adequar les interfícies per a tauletes i dispositius mòbils, encara que n'hi ha que ja ho estan.
- Fer les interfícies multiidioma.
- Possibilitat d'alternar les interfícies entre un tema clar i un tema fosc.
- Ampliació de l'aplicació per tal de fer-la més genèrica, de manera que pogués servir per a qualsevol facultat i universitat.
- Reescriure el codi font en TypeScript, per tal d'aprofitar les millores que presenta sobre el llenguatge actual, JavaScript.

Bibliografia

- [Comissió dels projectes final de grau de la UdG 2022] Comissió dels projectes final de grau de la UdG. *Guia dels projectes final de grau de l'enginyeria informàtica*, març de 2022. Disponible a https://moodle2.udg.edu/pluginfile.php/1779673/mod_resource/content/4/GuiaPFG_GEINF_amb_rubrica.pdf. (Citada a la pàgina 4.)
- [Contribuïdors de Sequelize 2022] Contribuïdors de Sequelize. *Sequelize - Feature-rich ORM for modern TypeScript and JavaScript*, juny de 2022. Disponible a <https://sequelize.org>. (Citada a la pàgina 72.)
- [Docker 2022] Docker. *Docker: Accelerated, Containerized Application Development*, juny de 2022. Disponible a <https://www.docker.com>. (Citada a les pàgines 73 and 135.)
- [Eduardo San Martin Morote 2022] Eduardo San Martin Morote. *Pinia*, juny de 2022. Disponible a <https://pinia.vuejs.org>. (Citada a la pàgina 68.)
- [Evan You, Eduardo San Martin Morote 2022] Evan You, Eduardo San Martin Morote. *Vue Router*, juny de 2022. Disponible a <https://router.vuejs.org>. (Citada a la pàgina 68.)
- [Evan You 2022a] Evan You. *Vite | Next Generational Frontend Tooling*, juny de 2022. Disponible a <https://vitejs.dev>. (Citada a la pàgina 65.)
- [Evan You 2022b] Evan You. *Vue.js | The Progressive JavaScript Framework*, juny de 2022. Disponible a <https://vuejs.org>. (Citada a la pàgina 66.)
- [GitHub 2022] GitHub. *GitHub*, març de 2022. Disponible a <https://github.com>. (Citada a la pàgina 64.)
- [Google 2022a] Google. *Angular*, juny de 2022. Disponible a <https://angular.io>. (Citada a la pàgina 66.)
- [Google 2022b] Google. *Chrome Dev Tools - Chrome Developers*, juny de 2022. Disponible a <https://developer.chrome.com/docs/devtools>. (Citada a la pàgina 70.)
- [Google 2022c] Google. *The color system - Material Design*, juny de 2022. Disponible a <https://m2.material.io/design/color/the-color-system.html>. (Citada a la pàgina 69.)

- [Google 2022d] Google. *Material Symbols and Icons - Google Fonts*, juny de 2022. Disponible a <https://fonts.google.com/icons?selected=Material+Icons>. (Citada a la pàgina 69.)
- [Hays 2022] Hays. *GUIA-DEL-MERCADO-LABORAL-2022-HAYS.pdf*, juny de 2022. Disponible a <https://www.legaltoday.com/wp-content/uploads/2022/01/GUIA-DEL-MERCADO-LABORAL-2022-HAYS.pdf>. (Citada a la pàgina 7.)
- [IETF 2022] IETF. *RFC 7519 - JSON Web Token (JWT)*, març de 2022. Disponible a <https://datatracker.ietf.org/doc/html/rfc7519>. (Citada a les pàgines 24, 78 and 109.)
- [Jeel Patel 2022] Jeel Patel. *Vue vs React: Which JavaScript framework is best?*, juny de 2022. Disponible a <https://www.monocubed.com/blog/vue-vs-react>. (Citada a la pàgina 66.)
- [JetBrains 2022] JetBrains. *DataGrip: The Cross-Platform IDE for Databases and SQL by JetBrains*, juliol de 2022. Disponible a <https://www.jetbrains.com/datagrip>. (Citada a les pàgines 74 and 87.)
- [JGraph Ltd 2022] JGraph Ltd. *Diagram Software and Flowchart Maker*, juny de 2022. Disponible a <https://www.diagrams.net>. (Citada a la pàgina 75.)
- [JSON 2022] JSON. *JSON*, desembre de 2022. Disponible a <https://www.json.org>. (Citada a la pàgina 21.)
- [LaTeX Project 2022] LaTeX Project. *LaTeX - A document preparation system*, juny de 2022. Disponible a <https://www.latex-project.org>. (Citada a la pàgina 75.)
- [Lokesh Gupta 2022] Lokesh Gupta. *REST Architectural Constraints*, desembre de 2022. Disponible a <https://restfulapi.net/rest-architectural-constraints>. (Citada a la pàgina 21.)
- [MDN Web Docs 2022] MDN Web Docs. *JavaScript | MDN*, desembre de 2022. Disponible a <https://developer.mozilla.org/docs/Web/JavaScript>. (Citada a la pàgina 19.)
- [Meta Platforms 2022] Meta Platforms. *React - A JavaScript library for building user interfaces*, juny de 2022. Disponible a <https://reactjs.org>. (Citada a la pàgina 66.)
- [Microsoft 2022] Microsoft. *Visual Studio Code - Code Editing. Redefined*, juny de 2022. Disponible a <https://code.visualstudio.com>. (Citada a la pàgina 74.)

- [npm 2022] npm. *npm*, març de 2022. Disponible a <https://www.npmjs.com>. (Citada a la pàgina 65.)
- [OpenJs Foundation 2022a] OpenJs Foundation. *Express - Node.js web application framework*, juny de 2022. Disponible a <https://expressjs.com>. (Citada a la pàgina 71.)
- [OpenJs Foundation 2022b] OpenJs Foundation. *Node.js*, juny de 2022. Disponible a <https://nodejs.org>. (Citada a les pàgines 71 and 136.)
- [Oracle 2022] Oracle. *MySQL*, juny de 2022. Disponible a <https://www.mysql.com>. (Citada a la pàgina 73.)
- [Postman 2022] Postman. *Postman API Platform*, juny de 2022. Disponible a <https://www.postman.com>. (Citada a la pàgina 72.)
- [Project Management Institute 2022] Project Management Institute. *Project Management Institute | PMI*, març de 2022. Disponible a <https://www.pmi.org>. (Citada a la pàgina 8.)
- [PULSARDEV SRL 2022] PULSARDEV SRL. *Quasar Framework*, juny de 2022. Disponible a <https://quasar.dev>. (Citada a la pàgina 69.)
- [Red Hat 2022] Red Hat. *What is a REST API?*, desembre de 2022. Disponible a <https://www.redhat.com/topics/api/what-is-a-rest-api>. (Citada a la pàgina 21.)
- [Scott Chacon, Jason Long, contribuïdors de Git 2022] Scott Chacon, Jason Long, contribuïdors de Git. *Git*, març de 2022. Disponible a <https://git-scm.com>. (Citada a la pàgina 63.)
- [TeamGantt 2022] TeamGantt. *Free Online Gantt Chart Maker That's Easy to Use | TeamGantt*, maig de 2022. Disponible a <https://www.teamgantt.com>. (Citada a la pàgina 75.)
- [W3C 2022a] W3C. *Cascading Style Sheets*, desembre de 2022. Disponible a <https://www.w3.org/Style/CSS>. (Citada a la pàgina 19.)
- [W3C 2022b] W3C. *HTTP - Hypertext Transfer Protocol*, desembre de 2022. Disponible a <https://www.w3.org/Protocols>. (Citada a la pàgina 21.)
- [WHATWG 2022] WHATWG. *HTML Standard*, desembre de 2022. Disponible a <https://html.spec.whatwg.org/multipage>. (Citada a la pàgina 19.)