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

Villagrasa Ramirez, Daniel; Grau Sala, Ramon, dir. Controlar nivells d'ansietat i depressió mitjançant una aplicació web per evitar l'abandonament universitari. 2022. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/264159>

under the terms of the  license

Controlar nivells d'ansietat i depressió mitjançant una aplicació web per evitar l'abandonament universitari

Daniel Villagrasa Ramírez

Resum– Aquest projecte vol ajudar a detectar valors d'ansietat i depressió dels alumnes, així com ajudar a relacionar la quantitat d'abandonaments amb aquests estats. Una vegada detectats els valors, la intenció es donar feedback per ajudar a l'alumne a millorar el seu estat si ho necessita i evitar que abandoni els estudis. El projecte es divideix en un entorn web acadèmic on es contesten els qüestionaris, una API Rest que gestiona les peticions web i consulta la base de dades a més d'enviar els correus amb feedback a l'alumne, una base de dades que desa les dades dels qüestionaris contestats i els expedients dels alumnes i finalment una eina amb la que es fa l'explotació de dades que permet visualitzar els resultats dels qüestionaris i agrupar per l'estat d'expedient dels alumnes, així com més filtres.

Paraules clau– Salut mental, abandonament universitari, desenvolupament web, API Rest, Software as a Service, explotació de dades

Abstract– This project aims to help detect values of anxiety and depression in students, as well as to help relate the number of dropouts to these conditions. Once the values have been detected, the intention is to give feedback to help students to improve their condition if necessary and prevent them from dropping out of school. The project is divided into an academic web environment where the tests are answered, an API Rest that handles web requests and retrieves data from the database as well as providing feedback via email to the student, a database that stores the data of the answered tests and the students' records, and finally a tool to extract and visualise data of the tests allowing to filter them according to the students' record status, among other filters.

Keywords– Mental health, university dropout, web development, API Rest, Software as a Service, data mining



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

PER posar en context, a Espanya, el suïcidi és la principal causa de mort no natural entre la població jove [1] i és només superada per malalties circulatories o respiratòries i els tumors com a principal causa de mort [2]. Per sort, la salut mental és un tema que cada vegada es tracta més a l'actualitat. S'ha impulsat la seva visualització al Congrés dels Diputats i les noves generacions trenquen el tabú parlant obertament de la seva experiència. A nivell acadèmic, un entorn de pressió i exigència com la

universitat pot fer que els alumnes acabin patint d'ansietat i altres trastorns relacionats, fins al punt d'obtenir males qualificacions o acabar deixant els estudis tal i com demostren Bethencourt et al [3] conclouent que les característiques psicològiques de l'alumnat tenen un gran impacte en la decisió de l'alumne d'abandonar els estudis.

Tenint en compte aquesta base també es pot aprofundir en l'estat dels alumnes, ja que aquests amb major propensió a abandonar els estudis presenten valors alts d'ansietat com a estat, ansietat com a tret i depressió, justament valors que són baixos en els alumnes que no abandonen [4]. A aquesta situació també se li ha d'afegir la pandèmia de la COVID, que ha servit per accentuar aquest tipus de problemes de forma més preocupant en els joves. En un informe de la OMS [5] es fa present que la salut mental dels joves s'ha vist més perjudicada després de la pandèmia de la COVID, essent més propensos a patir depressió, intents suïcides o

- E-mail de contacte: Daniel.Villagrasa@autonoma.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Ramón Grau Sala (Departament d'Arquitectura de Computadors i Sistemes Operatius)
- Curs 2021/22

d'auto-lesions.

Havent presentat aquestes dades, queda clara la problemàtica definida i una pregunta: Com podem evitar tendències depressives o d'ansietat en l'àmbit universitari i que aquestes acabin derivant en abandonament? La universitat es la casa de milers d'alumnes, majorment d'entre 18 i 25 anys, que es veuen sotmesos a situacions d'estrès i pressió constant. La resposta a l'anterior pregunta la dona l'article de Nuran Bayram & Nazan Bilgel [6] sobre els valors d'ansietat, estrès i depressió en els universitaris: s'ha de detectar aviat aquests estats, donar suport a l'alumne i que faci el tractament correcte.

Poder detectar indicis que puguin evitar que un alumne entri en una depressió o sigui víctima de l'ansietat hauria de ser una prioritat per a la universitat, no només per cuidar als seus alumnes, també perquè aquestes situacions també provoquen que l'alumne fracassi en els seus estudis i els abandoni.

1.1 Objectius

Així doncs, havent presentat ja el context, aquest TFG neix amb l'objectiu d'orientar a l'alumne en base al seu *feedback* per evitar l'abandonament escolar i una situació desagradable per a la seva salut mental. Aprofitant la meua estada a SIGMA AIE i el fet de que el seu servei *SaaS* és present a la majoria d'universitats catalanes, s'ha decidit desenvolupar un mòdul que funcionaria integrat al seu entorn, imitant el funcionament i aspecte però introduint els conceptes propis d'aquest projecte.

Els objectius principals del projecte són:

- Desenvolupar un entorn web per a l'estudiant on pugui accedir amb les mateixes credencials que al seu entorn universitari.
- Que en aquest entorn l'alumne pugui veure un menú desplegable on accedir a un mòdul orientat al seguiment psicològic per part de l'estudiant, on es calcularà el seu nivell d'ansietat com a estat, ansietat com a tret i de depressió.
- Realitzar una avaluació depenent de les respostes a cada formulari que anirà lligada a un feedback immediat a l'alumne que rebrà en forma de correu electrònic.
- Desar l'avaluació a la base de dades amb la data que s'ha fet i l'identificador de l'expedient de l'alumne que l'ha contestat per a poder saber l'estat actual i futur del seu expedient.
- Fer l'explotació de les dades desades per veure els nivells d'ansietat com a estat, ansietat com a tret i de depressió d'un conjunt d'alumnes filtrant per any i semestre acadèmics o pla d'estudis.
- Poder treure mètriques i gràfiques que relacionin l'estat mental d'un conjunt d'alumnes amb l'abandonament universitari.

1.2 Beneficis

Aquest projecte, degut al seu caire social, està orientat a aportar beneficis a la societat, tant als alumnes com a les universitats. Aquests beneficis són:

- Un entorn de confiança on els alumnes poden expressar el seu estat mental i emocional actual.
- Que a partir del seu feedback s'ofereixin solucions segons la seva valoració per aconseguir millorar el seu estat.
- Evitar l'abandonament universitari i millorar les qualificacions dels alumnes.
- Tallar d'arrel qualsevol índex de depressió, oferint ajuda mitjançant el servei assistencial de salut de la universitat en qüestió, el telèfon 024 o una cita amb un psicòleg si fos necessari.
- Aportar una forma de treure mètriques, pel que si un alumne abandona finalment un estudi es pugui saber si ha estat pels seus estats d'ansietat o depressió.
- Com es poden treure mètriques i informar de les causes d'abandonament tenint en compte la salut mental també es pot destinar més recursos a prevenir-les.

1.3 Estat de l'art

Que la salut mental és un aspecte a tenir en compte queda encara més clar en aquests últims anys. Durant la realització d'aquest TFG han començat a sortir estudis i articles que profunditzen en estats de depressió o ansietat focalitzats a l'àmbit escolar. Un d'aquests exemples el podem veure al projecte PsiCE [7], que vol realitzar un estudi a nivell estatal per poder prevenir problemes emocionals en estudiants. Tot i que aquest estudi està destinat a alumnes que cursen estudis secundaris i de batxillerat, és un primer pas important per conscienciar sobre la salut mental tant a docents com a estudiants. El PsiCE avaluarà a uns 14.000 estudiants de diferents Comunitats Autònomes mitjançant psicòlegs especialitzats.

A més d'aquest estudi també trobem mesures aprovades recentment, com la línia d'atenció a la conducta suïcida, estrenada el dia 9 de maig del 2022. El telèfon, el 024, es gratuït i atès per professionals. Durant el seu primer dia, aquest telèfon va rebre més de 1000 trucades, amb un temps d'atenció mitjana de 25 minuts [8]. El que vol fer aquest servei d'atenció es ajudar i evitar conductes que puguin portar a aquest desenllaç. Tractar el problema, fer que deixi de ser un tema tabú i que les persones amb problemes puguin parlar-ne i rebre un tractament és l'objectiu d'aquesta línia d'atenció.

A nivell acadèmic, però, no trobem una iniciativa semblant a la proposada o a l'estudi PsiCE. En quant a SIGMA, existeix un mòdul anomenat PAT (Pla d'Acció Tutorial). Aquest mòdul recau molt en la influència d'un tutor assignat, que ha de fer un seguiment a l'alumne interessant-se per ell i demanant opinions de com porta les assignatures. Busca un *feedback* més personalitzat i on la interacció és sempre des d'un tutor a un alumne, una tasca impossible si es vol fer amb tots els alumnes de la universitat. A més, el PAT s'orienta més a un seguiment tutoritzat de l'alumne a nivell acadèmic, ja que no tracta cap tema de salut mental. Tampoc es recopilen cap tipus de dades que puguin servir per a una explotació posterior.

El mòdul d'enquestes de SIGMA també té certa semblança al que proposa aquest treball, però una vegada

més les enquestes que els alumnes contesten estan destinades a rebre *feedback* de la càrrega de treball de les assignatures, com aquesta segueix les pautes definides a les guies docents o valorar el treball del professor. No es tracta en cap moment la salut mental ni busca per conèixer saber com es troba un alumne.

2 METODOLOGIA

En aquesta secció s'indica la metodologia que s'ha seguit per al desenvolupament del projecte. Inicialment s'ha previst un número de tasques i se'ls hi ha assignat un temps de desenvolupament estimat, tenint en compte que hi pot haver canvis tant en el número de tasques com en la duració de les mateixes. Aquestes tasques pertanyen a uns *sprints* i unes fases del desenvolupament del projecte.

S'ha treballat fent servir un desenvolupament àgil, utilitzant el *framework SCRUM* (adaptat a les característiques del treball) i *Kanban* per a l'estructuració i seguiment de les tasques generades. S'ha fet servir aquest *framework* per adaptar cada entrega d'informes considerant la data límit com la finalització d'un *sprint*. Per tant, s'han fixat unes tasques a realitzar i una entrega de producte per *sprint*, anant iterant sobre ella, afegint funcionalitats una vegada s'han provat i integrat correctament i rebent el *feedback* del tutor (en aquest cas com si fes el paper de *Stakeholder*) amb una reunió després de cada entrega. Tot i que no és una reproducció fidel de *SCRUM* y les seves bases, aquesta adaptació ha permès imitar el màxim possible un entorn de desenvolupament real.

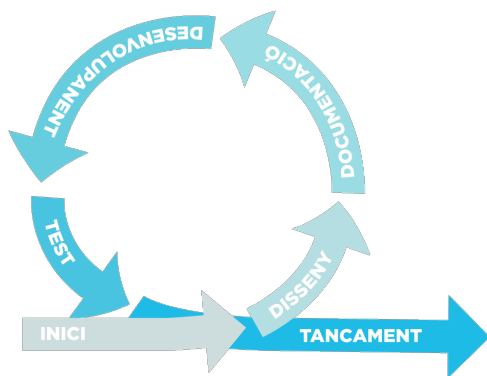


Fig. 1: Representació de la metodologia AGILE adaptada al projecte

S'han definit 4 *sprints*, coincidint amb les entregues dels diferents informes i que abasten diferents fases del projecte. Cada *sprint* estarà inclòs en diferents fases del projecte:

- Sprint 0: 14 de febrer del 2022 - 6 de març del 2022 (Inici de projecte i documentació)
- Sprint 1: 7 de març del 2022 - 10 d'abril del 2022 (Disseñy, desenvolupament i documentació)
- Sprint 2: 11 d'abril del 2022 - 22 de maig del 2022 (Desenvolupament, test, documentació i tancament del projecte)
- Sprint 3: 23 de maig del 2022 - 26 de juny del 2022 (Desenvolupament, test i tancament del projecte)

El detall de cada *sprint* es pot veure a l'apèndix A.2, abastant les diferents fases del projecte. A forma de resum, l'*sprint* 0 actua com a fase de preparació i documentació, l'*sprint* 1 serveix per acabar de definir els requisits dels qüestionaris i iniciar el desenvolupament web i de base de dades. L'*sprint* 2, per la seva part, es centra completament en el desenvolupament de la *API Rest* i una inicialització a *Power BI*, que es finalitza a l'*sprint* 3 amb la creació dels informes d'exploració de dades i els informes finals del projecte.

Pel seguiment de les tasques a realitzar s'ha utilitzat el mètode *Kanban* mitjançant un tauler de *Trello*. Tal y com es pot veure a la figura 2 s'han establert diferents columnes que representen els estats en que es pot trobar la tasca. Al *backlog* tenim tasques possibles que poden aprofundir i millorar el projecte, a *to-do* les tasques del projecte que falten per fer, a *doing* les que estan en desenvolupament, a *test* les que s'estan provant abans d'implementar definitivament i a *done* les que ja formen part d'una versió estable.

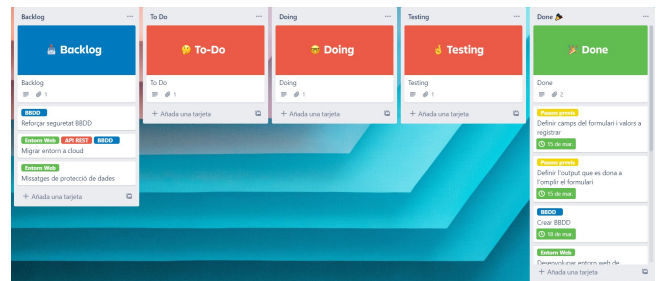


Fig. 2: Tauler de Trello del projecte

3 ELICITACIÓ DE REQUISITS

En aquest apartat es defineixen els requisits del sistema, així com les restriccions, basant-se en els objectius definits a l'apartat 1.1 i els casos d'ús que es mostren a l'apartat 3.4. Els requisits funcionals defineixen les funcionalitats desitjades amb les que comptarà el producte. D'altra banda, també es mostren els requisits no funcionals, que indiquen característiques del funcionament del producte i restriccions que apliquem al seu desenvolupament.

3.1 Requisits funcionals

A la taula 1 es presenten els requisits funcionals del sistema.

3.2 Requisits no funcionals i restriccions

A la taula 2 trobem els requisits no funcionals i les restriccions del sistema.

3.3 Perspectiva del producte

La figura 3 mostra l'arquitectura del sistema, on es poden veure les diferents parts que el formen i les relacions entre elles. Així es pot veure de forma gràfica la perspectiva del nostre producte.

ID	Descripció
RF-001	El sistema ha de permetre a l'usuari alumne que inici sessió al web
RF-002	El sistema ha de mostrar a l'usuari alumne l'opció de contestar qüestionaris pendents
RF-003	El sistema ha de mostrar a l'usuari alumne l'opció de veure els qüestionaris contestats
RF-004	El sistema ha de mostrar a l'usuari alumne els qüestionaris pendents de contestar
RF-005	El sistema ha de permetre a l'usuari alumne contestar les preguntes del qüestionari seleccionat
RF-006	El sistema ha de permetre a l'usuari alumne enviar el qüestionari
RF-007	El sistema ha de donar feedback a l'usuari alumne una vegada ha contestat el qüestionari
RF-008	El sistema ha de permetre a l'usuari alumne tancar sessió al web
RF-009	El sistema Power BI ha de permetre a l'usuari de gestió acadèmica consultar informes de dades
RF-010	El sistema Power BI ha de permetre a l'usuari de gestió acadèmica aplicar filtres als informes
RF-011	El sistema Power BI ha de permetre a l'usuari de gestió acadèmica exportar informes generats

TAULA 1: REQUISITS FUNCIONALS DEL SISTEMA

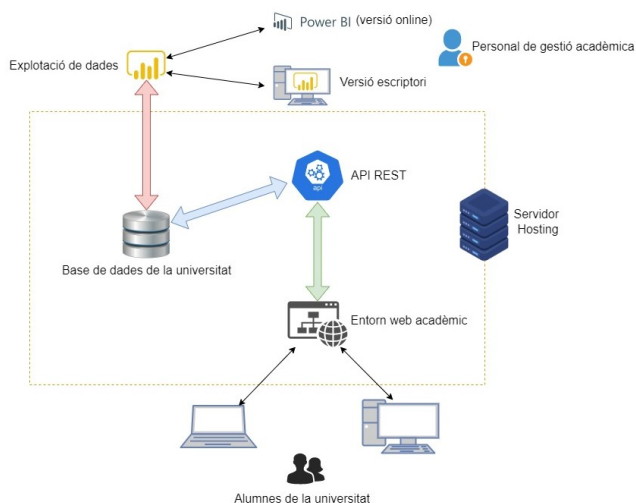


Fig. 3: Arquitectura del sistema

3.4 Usuaris i casos d'ús

Aquest projecte està orientat a que l'utilitzin 2 tipus d'usuaris, simplificant l'estil que SIGMA AIE utilitza en el seu entorn. Els usuaris del sistema i les seves funcions, són:

- Alumne: Usuari alumne de la universitat que utilitzarà l'entorn web i contestarà els qüestionaris. També és a

ID	Descripció
RNF-001	El sistema es desenvoluparà per a utilitzar-se des d'un navegador web
RNF-002	El sistema ha de comptar amb un servidor local per allotjar la web, una API i una base de dades
RNF-003	El sistema comptarà amb els qüestionaris d'ansietat-estat, ansietat-tret i depressió
RNF-004	El sistema permetrà contestar a un alumne un tipus de qüestionari per semestre i any acadèmic
RNF-005	El sistema no mostrarà un qüestionari si aquest ja s'ha contestat al mateix semestre i any acadèmic
RNF-006	El sistema ha de tenir una interfície d'usuari seguint l'estil de SIGMA AIE
RNF-007	El sistema demanarà a l'usuari alumne que s'identifiqui amb les seves credencials
RNF-008	El sistema imitarà l'estil de la BBDD de SIGMA AIE
RNF-009	El sistema avisarà a l'usuari si s'ha deixat alguna pregunta per contestar després de donar al botó d'enviar
REST-001	El back-end del sistema es desenvoluparà seguint el framework Spring
REST-002	El back-end del sistema utilitzarà la versió 11 de Java
REST-003	El back-end del sistema es desenvoluparà amb la última versió del IDE Eclipse
REST-004	El back-end del sistema enviarà correus electrònics fent servir la API de Gmail
REST-005	El front-end del sistema es desenvoluparà seguint el model vista controlador
REST-006	El front-end del sistema utilitzarà la versió 7.0 de PHP
REST-007	El front-end del sistema utilitzarà llibreries de bootstrap pels dissenys
REST-008	La base de dades del sistema serà MySQL
REST-009	La base de dades, front-end i back-end s'executaran en local mitjançant XAMPP i PhpMyAdmin

TAULA 2: REQUISITS NO FUNCIONALS I RESTRICCIONS DEL SISTEMA

qui va dirigit el *feedback* resultat del qüestionari contestat.

- Gestió Acadèmica: Usuari que podrà accedir a l'entorn d'explotació de dades (*Power BI*) i podrà consultar i exportar els diferents informes disponibles.

Una vegada definits els usuaris del sistema, la figura 4

mostra els diferents casos d'ús que els usuaris portaran a terme al sistema.

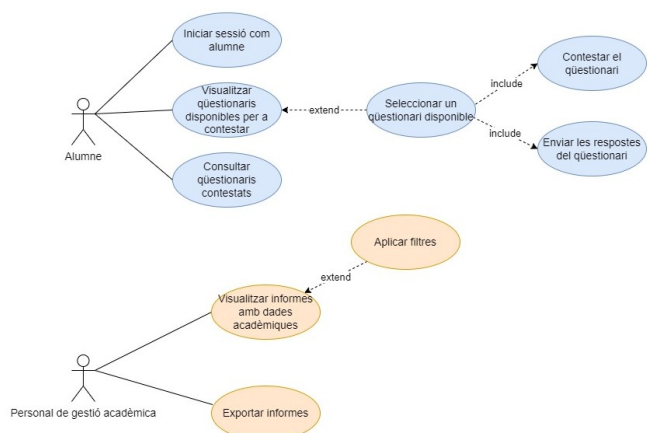


Fig. 4: Casos d'ús

4 CERCA DE QÜESTIONARIS

Finalment, abans de passar a parlar del desenvolupament del sistema, aquest apartat vol aprofundir en una part vital del projecte: els qüestionaris que volen recollir els diferents valors d'ansietat i depressió dels alumnes. Degut a que aquest és un camp delicat i del que no compto amb la formació suficient com per a tenir un criteri vàlid, he demanat ajuda d'un company psicòleg de la *Universidad de Jaén*. La intenció d'aquest projecte sempre ha estat conèixer dos característiques dels alumnes: el nivell d'ansietat i si es troben en un estat depressiu. Finalment, els qüestionaris escollits han estat dos (*STAI* i *BDI*), tot i que un d'ells es divideix en dos diferents. A continuació s'expliquen els qüestionaris, l'*output* que dona l'aplicació i com es tracten els resultats.

4.1 Ansietat - STAI

Per definir els nivells d'ansietat de l'alumne s'utilitzarà el *State-Trait Anxiety Inventory* (STAI) amb les seves variants A/E (Ansietat com Estat) [11] i A/R (Ansietat com a Tret) [12]. Els dos test s'utilitzen per trobar valors en aquest mateix moment, pel que es poden repetir cada cert temps amb resultats diferents segons l'estat actual de la persona.

- Ansietat com Estat (STAI A/E): Aquest qüestionari avalua un estat emocional transitori, caracteritzat per sentiments subjectius d'atenció i aprensió així com hiperactivitat del sistema nerviós.
- Ansietat com a Tret (STAI A/R): D'altra banda, aquest qüestionari busca assenyalar una propensió ansiosa característica d'individus amb tendència a percebre les situacions com a amenaçadores.

Els dos qüestionaris es contesten de la mateixa forma per part de l'usuari: ha d'assignar un valor de 0 a 3 a una pregunta concreta, amb valors (Res, una mica, bastant i molt) respectivament.

4.1.1 Avaluació i ponderació

Per a obtenir el resultat de cada test es fan servir unes plantilles que donen una puntuació segons la resposta donada per l'usuari. Per a no falsificar respostes per la naturalesa dels valors, hi ha unes preguntes que compten amb valors invertits, pel que si un usuari contesta amb una resposta que en condicions normals tindria un valor 0, en aquestes acaba sumant 3. La fórmula per a calcular els resultats és:

- A/E: $30 + \text{valors normals} - \text{valors negats}$
- A/R: $21 + \text{valors normals} - \text{valors negats}$

Una vegada obtinguts els resultats, s'ha de definir el llinar on es troba. A partir de la ponderació oficial, aquesta s'ha adaptat per al projecte considerant només la franja d'edat de joves entre 18-25 anys i el gènere masculí. La ponderació és doncs la següent:

	A/E	A/R
Alt	29 - 60	26 - 60
Sobre mitjana	20 - 28	20 - 25
Mitjana	19	19
Tendència mitjana	14 - 18	14 - 18

TAULA 3: PONDERACIÓ DELS QÜESTIONARIS STAI

4.1.2 Definició de feedback per part de l'aplicació

Després d'obtenir un valor, el següent pas es donar un *feedback* a l'alumne en base a l'avaluació realitzada.

- Ansietat alta: Correu on s'avis de valors alts de la mitjana. Es vol conscienciar a l'alumne de que demani ajuda professional mitjançant el servei de salut de la universitat, fent-li arribar el contacte amb ells, així com suggerir que busqui ajuda per altres canals com els de la Seguretat Social o psicòlegs privats. A nivell acadèmic, es recorda a l'alumne que si se sent desbordat per la quantitat d'assignatures, pot matricular menys assignatures si així se sent més còmode i que si té problemes amb assignatures actuals demani tutories als professors.
- Ansietat sobre mitjana: Correu on s'avis de valors una mica més alts de la mitjana. Es transmet a l'alumne que recordi que pot matricular menys assignatures si així se sent més còmode, que si té problemes amb assignatures actuals demani tutories als professors i que si necessita ajuda professional la pot trobar al servei de salut de la universitat, fent-li arribar el contacte amb ells.
- Ansietat mitjana: Correu amb avís de valoració mitjana i recordant que tot i així faci el test el següent semestre i que si necessita assistència contacti amb el servei assistencial de salut de la universitat.
- Ansietat amb tendència mitjana: Correu amb avís de valoració mitjana i recordant que tot i així faci el test el següent semestre.

4.2 Depressió - BDI-II

Per definir si un alumne té indicis de depressió o no, l'alumne realitzarà el test *Beck Depression Inventory* (BDI-II) [13] on descriurà com s'ha sentit les últimes 2 setmanes. Aquest test està destinat a calcular la gravetat de nivell depressiu d'un adult a partir de 13 anys. Consta de només una variant de 20 situacions descrites que es poden contestar amb 4 possibles respostes per cadascuna.

4.2.1 Avaluació i ponderació

En aquest cas la forma d'avaluar els resultats és més simple. Cada opció consta de 4 respostes possibles amb una puntuació de 0 a 3, sense valors negats. Al final es fa un sumatori dels resultats. Una vegada obtinguts els valors, s'ha de definir què representa el valor obtingut per l'usuari. Els diferents valors possibles, són:

	BDI - II
Depressió severa	30 - 63
Depressió moderada	17 - 29
Depressió lleu	10 - 16
Depressió mínima	14 - 18

TAULA 4: PONDERACIÓ DEL QÜESTIONARI BDI-II

4.2.2 Definició de feedback per part de l'aplicació

Després d'obtenir un valor, el següent pas és donar un *feedback* a l'alumne en base a l'avaluació realitzada.

- **Depressió severa:** Segurament es tracti d'un alumne que ja compta amb atenció especialitzada. S'envia un correu on es recomana ajuda professional si no compta amb ella, fent arribar el contacte amb el servei de la universitat, també es suggereix que busqui ajuda per altres canals com els de la Seguretat Social o psicòlegs privats si no ho ha fet encara.
- **Depressió moderada:** S'envia un correu on es recomana ajuda professional, fent arribar el contacte amb el servei de la universitat.
- **Depressió lleu:** Correu amb avís de valoració mitjana i recordant que tot i així faci el test el següent semestre i que si necessita assistència contacti amb el servei assistencial de salut de la universitat.
- **Depressió mínima:** Correu amb avís de valoració mínima i recordant que tot i així faci el test el següent semestre.

5 DESENVOLUPAMENT

En aquesta secció s'indicarà com s'han desenvolupat les 4 parts que formen el projecte, comentant el seu funcionament, esquelet i com interaccionen entre elles. També s'utilitzaran materials presents a l'apèndix per aportar una explicació més detallada.

5.1 Entorn web

Pel desenvolupament de l'entorn web s'ha fet servir el Model-Vista-Controlador [9] utilitzant com a llenguatges *PHP*, *HMTL*, *Javascript* i *CSS*. L'*IDE* per al seu desenvolupament ha estat *PhpStorm*. Cada pàgina que es visualitza a l'entorn web consta d'una vista que mostra un contingut que gestionen els controladors ja sigui mitjançant càlculs interns o consultes fetes -mitjançant el model- a la *API*.

El funcionament de l'entorn web comença al fitxer *index.php*, que es pot veure a l'apèndix A.3. A partir d'aquest arxiu es defineixen casos, on cada cas serà una *URL* a seguir. Per cada cas es farà una crida a uns *resources*, que són un esquelet de crides a arxius de *Javascript*, estils i altres parts de la pàgina web, aquests *resources* redirigeixen a un controlador on s'agafen les variables i es fan les crides a la *API* necessàries per mostrar contingut a les vistes. Es pot trobar un exemple d'aquests arxius de *resources* a l'apèndix A.4

Les crides a la *API* es fan al model mitjançant *CURL*, definint la *URLbase* on es troba la nostra *API Rest* escoltant. Aquest fitxer, que és l'únic que considerem al model, es divideix entre la crida a la *API* amb les diferents opcions que es poden utilitzar (*POST*, *PUT*, *DELETE*) i cada funció cridada pels controladors que indiquen la opció a utilitzar, els paràmetres a passar i la *URL* a seguir, com es pot veure en detall a l'apèndix A.5

El controlador rep els resultats una vegada fetes les crides al model o a funcions de *JavaScript* i *PHP*, per exemple per fer el càlcul dels resultats dels qüestionaris una vegada s'han respost o per recollir la informació a enviar en els correus. La vista, per la seva part, només fa la funció de mostrar per pantalla allò que li indiquem tant de forma directa al codi (com els qüestionaris) o el contingut que rebem després de consultar la *API*, utilitzant variables de *PHP*.

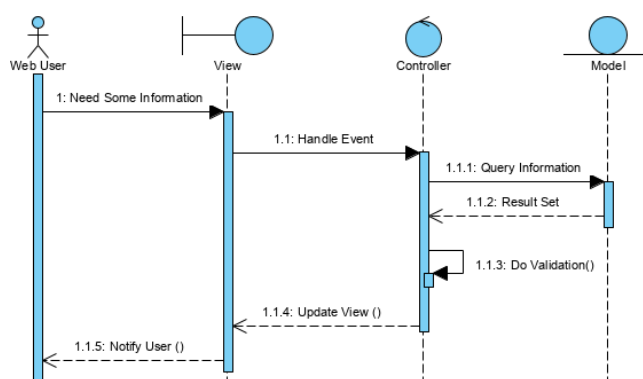


Fig. 5: Diagrama de seqüència que mostra un exemple del funcionament del Model-Vista-Controlador [14]

En quant al funcionament de la característica principal del programa, aquesta comença una vegada l'alumne veu els qüestionaris disponibles que té per a contestar, selecciona un, el respon i decideix enviar-lo. Tal i com es pot veure al codi que es mostra a l'Apèndix A.6, la informació del qüestionari contestat passa a uns controladors (que depenen del tipus de qüestionari) on es calculen mitjançant funcions de *PHP* el resultat seguint les ponderacions definides als apartats 4.1.1 i 4.2.1. Aquesta valoració es parametrilitza juntament amb l'identificador de l'expedient de l'alumne, l'any i semestre acadèmic i el tipus de qüestionari contes-

tat, es passa a format *JSON* i s'envia a la *API REST* per a que aquesta ho emmagatzemi a la base de dades a la taula *cuestionarioexpediente*.

Aquí no acaba el procés, ja que paral·lelament a la instrucció de desar a la base de dades la informació, també es crida a una funció de *Javascript* (Apèndix A.7) que passa a un controlador la informació sobre el tipus de qüestionari contestat i la valoració atorgada. Segons aquestes dues dades es tria el missatge adequat i es crida una vegada més a la *API* per a llançar el procés d'enviament de l'*email* a l'alumne, que es pot veure en la versió resumida del controlador *enviaMailAE.php* a l'apèndix A.8.

Finalment, s'ha tingut en compte que un alumne no pugui contestar un qüestionari més d'una vegada per semestre. S'ha desenvolupat un mecanisme de control que recull l'any i el semestre en els que ens trobem, de forma que quan un alumne contesta els qüestionaris, aquests es pugen a la base de dades amb una data real. Aquestes dates permeten fer aquest control, tornant-los a deixar disponibles quan s'inicia un nou semestre.

5.2 API Rest

Per a gestionar les peticions que es fan des de l'entorn web s'ha desenvolupat una *API REST* fent servir *Spring Boot* i la seva aproximació al Model-Vista-Controlador, com es pot veure a la figura 6, seguint la documentació proporcionada per la pròpia web de *Spring* [10]. La *API*, desenvolupada amb *Java* i l'*IDE Eclipse*, s'executa en local i escolta a través del port 8080. Des de l'entorn web s'envien peticions *POST* o *GET* que la *API* gestiona. Aquestes peticions estan relacionades amb el retornar una informació demanada, com per exemple dades dels usuaris, expedients d'alumnes o dades dels qüestionaris.

Gràcies al *framework* de *Spring Boot* resulta senzill realitzar les operacions de base de dades que necessita el nostre entorn com els *SELECT* o *INSERT*, per tant era l'opció adient per al desenvolupament de la *API*. Les peticions d'inserció d'informació estan relacionades amb el fet de gravar un nou qüestionari contestat per un alumne o de realitzar l'enviament d'un correu electrònic a un alumne donant *feedback* del seu resultat.

Els fitxers que conformen aquesta *API* es divideixen entre repositoris, models, controladors i un *main*. Per cada taula de la nostra base de dades es crearà un model, controlador i repositori.

Als controladors de la *API* s'escolta tota l'estona. Quan entra una petició per una *URL* definida, aquesta es tracta segons la funció desenvolupada al controlador, que farà una crida al repositori de funcions de *Spring Boot* (tant definides per defecte que fan operacions habituals *SQL* com aquelles que nosaltres mateixos hem pogut definir expressament). Aquestes consultes s'executen a la base de dades gràcies als arxius de model, que mapegen els atributs a la nostra *API*, respectant les relacions (que també s'han d'indicar) i establint la connexió.

Un exemple d'aquest funcionament es pot demostrar perfectament quan es fa una petició d'inserció a la base de dades amb un qüestionari contestat per un alumne. Inicialment el controlador espera una petició a la *URL* que hem definit i a la que el *front-end* (la nostra web) haurà cridat

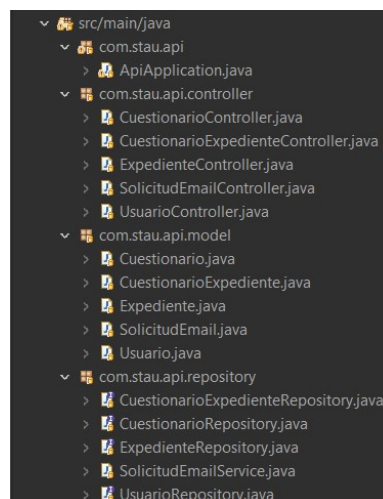


Fig. 6: Organització dels fitxers de la *API Rest* mostrant el model de *Spring*

fent una comanda *POST*. Aquesta crida que rep el controlador la podem trobar a l'Apèndix A.9

Al model s'hauran definit els camps de la taula implicada, així com les relacions amb les taules (claus secundàries, restriccions si n'hagués i claus primàries), mapejant tota la informació. També es defineixen els *getters* i *setters* que permeten obtenir i desar dades a la base de dades, que podem veure a l'Apèndix A.10.

Seguint amb el procés, el controlador farà una crida a una funció de *Spring* que forma part del *CRUD Reposirotiy* anomenada '*save*' que transforma la nostra petició en una sentència *INSERT*. Si les funcions que es criden des del controlador no estan definides a aquest repositori de *Spring*, es poden definir als diferents repositoris del projecte. Un exemple es la funció '*findBy*' que fa un *SELECT* a la taula *CuestionarioExpediente* del camp que li indiquem a continuació (*Idexp* en aquest cas). També es troben versions complexes que requereixen de la redacció d'una sentència *SQL* completa, ja que el que es demana fer no ho acaba d'automatitzar el *framework* de *Spring*. En el cas d'aquest projecte només s'ha hagut de fer una vegada i perquè es necessitava fer un '*Group By*' per filtrar els qüestionaris contestats per un alumne concret en un semestre i any acadèmic. Totes dues funcions i un exemple del funcionament dels arxius repositori es poden trobar en detall a l'Apèndix A.11.

A més de les operacions que s'ha mostrat, també s'utilitza la *API* per enviar un *e-mail* a l'alumne amb la valoració calculada del *front-end*. D'aquesta forma el que es rep es una petició *POST* amb una *array* de dades *JSON*, que es gestiona seguint el mateix esquema però canviant el concepte de repositori per utilitzar un servei. Per a realitzar aquest procés s'ha seguit el tutorial de Hamid Shaikh [15], adaptant completament el material a les necessitats del projecte. També s'ha creat un compte de *Gmail* des del que es realitza l'enviament dels correus electrònics. Una versió simplificada del servei es pot trobar a l'apèndix A.12.

5.3 Base de dades

La base de dades del projecte també s'allotja de forma local. S'ha creat utilitzant *phpmyadmin*, es gestiona amb el programa *XAMPP* i el seu llenguatge es *mySQL*. A l'apèndix

A.1 es pot veure el diagrama de classes de la base de dades per tenir una idea de les taules i les relacions. Tot i així, en aquest apartat es comentarà què emmagatzema cada taula:

- *usuario*: Aquesta taula guarda les dades d'un usuari com el seu nom i cognoms, correu i contrasenya. Els hi assigna un identificador.
- *expediente*: Aquesta taula es genera per cada usuari alumne que cursa un pla d'estudis. Conté informació rellevant de l'expedient acadèmic com l'estat (obert o tancat) i el motiu (cursant, abandonament o titulat)
- *cuestionario*: Conté l'identificador i el nom dels 3 tipus de qüestionaris.
- *cuestionarioexpediente*: Taula que neix de la unió d'*expediente* i *cuestionario*. Cada cop que un alumne contesta a un qüestionari es genera una entrada en aquesta taula que conté un identificador propi, així com el del qüestionari contestat i el de l'expedient de l'alumne. També conté les dates (semestre i any) on es contesta i la valoració segons la ponderació explicada a la secció 4.

5.4 Entorn d'exploració de dades

Per a tenir registres sobre els que treballar i treure mètriques, s'ha generat un conjunt de dades de prova per a omplir la base de dades de registres dels que poder extreure mètriques, imitant el funcionament real de l'aplicació. Òbviament, en un escenari real, s'hauria d'esperar a que els alumnes contestin els qüestionaris i passat uns anys treure les mètriques. Per a provar el funcionament i el concepte del treball s'ha decidit treballar amb aquestes dades "dummy" que han demostrat ser útils pel que es van idear.

S'ha decidit utilitzar el programa de *Microsoft Power BI*, que permet obtenir fàcilment les dades de la base de dades amb la que s'ha relacionat i poder generar tot tipus de gràfiques. També permet generar plantilles filtrant mètriques específiques i definides, de forma que aquestes mostren gràficament la informació desitjada.

En aquest cas el desenvolupament no ha estat lligat a la programació sinó a aprendre a utilitzar aquesta eina i crear de forma gràfica les diferents pantalles (anomenades informes) que, una vegada generades, seran accessibles per als usuaris destí. Així doncs, el treball important d'aquesta part del projecte ha estat assegurar que els informes aportin valor i compleixin els objectius definits a l'inici del projecte. Per tant, es decideix crear 3 informes:

- Inici
- Resultats d'enquestes
- Estat expedients per qüestionari

L'informe d'inici està pensat per mostrar-se com a pàgina inicial de l'entorn. Aquest informe consta d'un comptador amb el total del número d'expedients que va canviant segons els filtres aplicats. També es troba un gràfic que mostra l'estat dels expedients consultats i el número de qüestionaris que s'han contestat per cada tipus. Finalment, els filtres disponibles ens permeten filtrar per l'any acadèmic

de la última matrícula realitzada o pel codi del pla docent. Selecció dels valors dels filtres els resultats en pantalla s'adapten. Es pot consultar a la figura 9.

A l'informe de resultat d'enquestes es mostra només un gràfic de barres. El que es demana es que l'encarregat de gestió acadèmica que consulti aquest apartat comenci a triar els diferents filtres. El filtre de qüestionari permet veure els resultats segons el tipus de qüestionari contestat, que és la primera selecció a tenir en compte per veure uns resultats útils. Després trobem dos filtres que permeten aprofundir amb més detall al que es vol veure. Aquests filtres són el de l'últim any matriculat i el del codi del pla docent.

Finalment, l'estat d'expedients per qüestionari és l'informe més rellevant, ja que el que mostra és el gràfic de barres que s'ha vist anteriorment, però incloent també l'estat dels expedients dels alumnes segons els qüestionaris contestats. Per tant, el primer a fer és seleccionar un tipus de qüestionari pel qual es volen filtrar els resultats. Després es mostrarà al gràfic els resultats d'aquest qüestionari indicant també els estats dels expedients. Igual que als informes anteriors es pot aprofundir més en detall filtrant per l'últim any de matrícula o el pla docent. El gràfic resultant ens permet veure amb molt detall si hi ha relació entre els estats d'ansietat o depressió i si un alumne abandona la seva carrera o no. Es pot veure el gràfic resultant a la figura 10. Els informes generats es poden accedir per part del personal de gestió acadèmica mitjançant l'entorn online de *Microsoft* o la pròpia aplicació local de *Power BI*.

6 RESULTATS

Com a resultat del procés de desenvolupament, s'ha acabat obtenint un entorn web que com es pot veure a la figura 7 mostra a un alumne els qüestionaris que té disponibles per a contestar en el semestre i any actual. Una vegada l'usuari accedeix a un dels qüestionaris disponibles se li mostren les preguntes com a la figura 8, de forma que els pot contestar segons les seves preferències i després enviar-lo. Seguidament rebrà al seu correu un missatge amb el *feedback* de forma immediata. Aquest *feedback* va lligat a la ponderació definida pel qüestionari contestat.

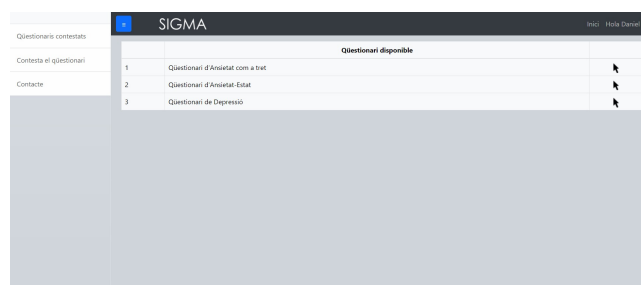


Fig. 7: Pantalla de l'entorn web del projecte de l'apartat de qüestionaris disponibles

D'altra banda, aquestes dades es podran visualitzar des de l'entorn d'exploració de dades gràcies als informes generats. Com es pot veure a la figura 9 es mostra un panell de control amb diferents filtres disponibles i mètriques. També es pot accedir a altres informes que mostren dades amb major detall, com l'informe que mostra els resultats de les enquestes o el que relaciona l'estat dels expedients segons el

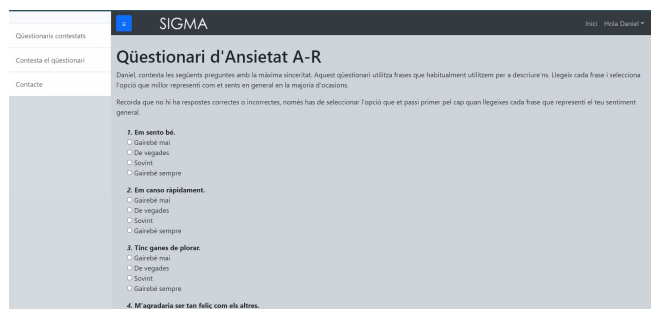


Fig. 8: Pantalla de l'entorn web del projecte que mostra un qüestionari a contestar

qüestionari contestat i la valoració obtinguda (figura 11).

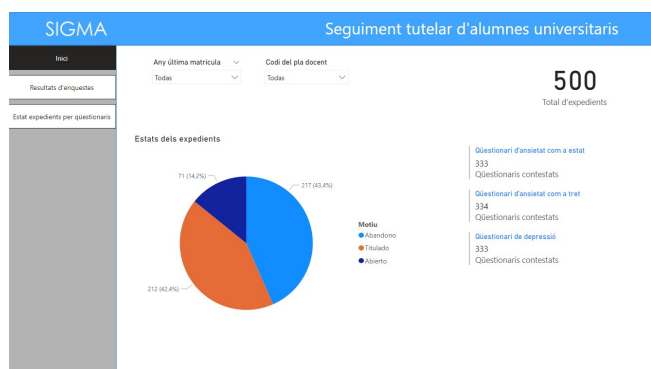


Fig. 9: Informe 'Inici' creat a Power BI

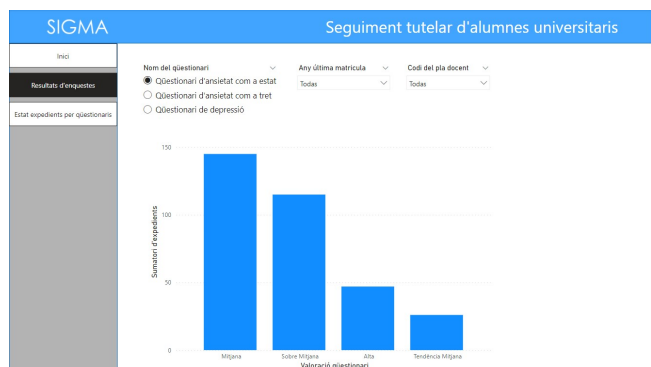


Fig. 10: Informe 'Resultats d'enquestes' creat a Power BI

Per la part del *back-end* també s'ha de destacar que s'ha obtingut una *API Rest* que fa les funcionalitats desitjades, que rep i gestiona de forma correcta les peticions que li envien.

7 CONCLUSIONS

Després d'explicar tot el desenvolupament i visualitzar els resultats, es poden extreure varies conclusions. La primera és que es pot afirmar que els objectius del projecte s'han complert en quant a les funcionalitats proposades. El programa per la part de l'usuari alumne recull les dades de forma correcta, a més de donar *feedback* de la seva situació i ajudar a la seva salut mental. L'explotació de dades demostra que amb la definició de classes i relacions de la base de dades es poden extreure gràfics potents que serveixen per

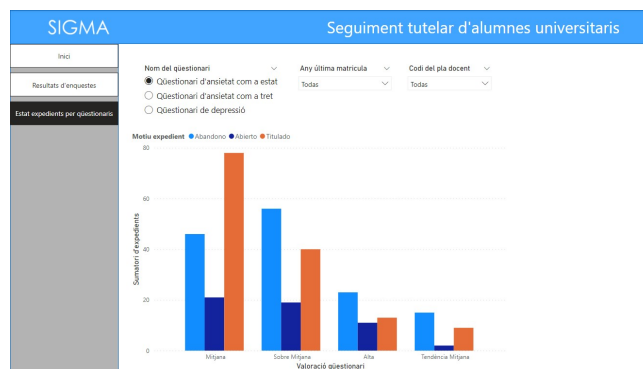


Fig. 11: Informe 'Estat expedients per qüestionari' creat a Power BI

conèixer dades com l'estat d'ansietat i depressió dels alumnes, filtrant per pla d'estudis, anys, semestres, etc.

La utilització de *Power BI* ha estat un encert, ja que permet creuar totes les dades que es considerin importants. Si seguim el que s'ha definit als objectius del projecte, podem extreure correctament mètriques que relacionin l'abandonament escolar amb els nivells d'ansietat i depressió d'un conjunt d'estudiants i el moment en que van abandonar. També, si els estudiants han anat contestant els qüestionaris al llarg dels seus estudis es pot fer un seguiment dels nivells d'ansietat i depressió durant aquest període. Així doncs, es pot assegurar que s'assoleix correctament aquesta part del projecte.

En quant al desenvolupament del projecte, s'ha demostrat que s'ha pogut adaptar als canvis, sense impactar de forma negativa el temps de desenvolupament. A més, s'ha pogut introduir noves tecnologies que inicialment no s'havien tingut presents, com *Power BI*.

7.1 Línies de millora

Durant el desenvolupament també s'ha pensat en funcionalitats extra que es poden afegir en el futur per evolucionar el concepte d'aquest i millorar-lo o adaptar-lo per quan es trobi en un entorn real o porti uns anys en funcionament:

- Tenir en compte el gènere per a millorar la detecció dels qüestionaris: La ponderació oficial dels resultats dels qüestionaris varia lleugerament depenent del gènere de l'alumne. Actualment no es té en compte a l'avaluació, però es podria afegir. Així doncs, es pot agafar aquest valor i calcular el resultat tenint-lo en compte. També s'hauria d'estudiar la opció de gènere no binari.
- Traspàs de l'entorn a un servidor online: Després del desenvolupament en *localhost*, si el projecte vol operar de forma online necessita un servidor allotjat a internet. Ja que aquest projecte s'ha desenvolupat a SIGMA, la intenció es fer la migració als seus servidors. Si fos en cas particular d'una universitat, s'hauria de migrar a l'entorn propi i els seus servidors.
- Millorar la protecció de dades: Oferir a l'alumne opcions per personalitzar el tractament de les dades, de forma que pot seleccionar quines dades vol que el sistema reculli, donant la opció de que no es relacionin

els seus resultats amb el seu expedient, tot i que conti-
nuï rebent feedback per cada qüestionari contestat.

- Tests més exhaustius: La part de test del projecte s'ha limitat a comprovar que la integració i el funcionament són correctes però no s'han revisat condicions de se-
guretat molt específiques. En el futur es poden provar
més casos i assolir millor nivell de seguretat.
- Entrenar models de predicció en base a resultats: Amb
dades reals, es poden entrenar models per detectar si
un estudiant es troba en perill d'abandonar els seus es-
tudis segons els seus estats d'ansietat i depressió. Si
es detecta, es podria fer un seguiment més personalit-
zat per encara intentar amb major èxit que no abandoni
els seus estudis. Aquesta part també va lligada amb la
protecció de dades i opcions que es volen oferir.
- Provar la hipòtesi que relaciona abandonaments amb
estat depressiu o d'ansietat: Amb dades reals i multi-
tud de resultats es pot fer un estudi que acabi provant
si les hipòtesis plantejades en els articles citats i la mo-
tivació del projecte són correctes.

AGRAÏMENTS

Agrair al Ramon Grau, qui ha estat sempre disponible per
guiar-me durant la realització del projecte amb sugger-
iments molt interessants que han ajudat a millorar el resultat
final. A nivell personal voldria agrair a la Rosa, la meua
psicòloga, per ajudar-me a ser qui soc i sempre creure en
mi; a Alba per ser un model a seguir i la millor compa-
nyia de vida que podria imaginar; a la Victoria, Zakariae i
tots els companys de SIGMA amb qui he après tant aquest
últim any i m'han fet sentir com a casa des del primer dia;
també a l'Antonio per ajudar-me amb els qüestionaris del
projecte. Finalment, a la meua família per la paciència i el
suport.

REFERÈNCIES

- [1] El Heraldo, «El suicidio se convier-
te en la primera causa de muerte en-
tre los jóvenes,» [En línea]. Available:
[https://www.heraldo.es/noticias/nacional/2021/07/25/
el-suicidio-se-convierte-en-la-primera-causa-de-
muerte-entre-los-jovenes-1508739.html](https://www.heraldo.es/noticias/nacional/2021/07/25/el-suicidio-se-convierte-en-la-primera-causa-de-muerte-entre-los-jovenes-1508739.html)
- [2] Huffington Post, «La pandemia silenciosa: el suici-
dio en España, en siete gráficos,» [En línea]. Avai-
lable: [https://www.huffingtonpost.es/entry/graficos-
suicidio-espana_es_61b85b8de4b0358b5c4af1c5](https://www.huffingtonpost.es/entry/graficos-suicidio-espana_es_61b85b8de4b0358b5c4af1c5)
- [3] J. T. Bethencourt Benítez, L. Cabrera Pérez, J. A.
Hernández Cabrera, P. Álvarez Pérez y M. González
Afonso, «Variables psicológicas y educativas en el
abandono universitario,» Revista electrónica de In-
vestigación Psicoeducativa, vol. 6, n° 16, pp. 603-622,
2008.
- [4] Caballero-Dominguez, C., Gallo-Barrera, Y., Suárez-
Colorado, Y. (2018). «Algunas variables de salud
mental asociadas con la propensión al abandono de los
estudios universitarios». *Psychologia*, 12 (2), 37-46.
doi: 10.21500/19002386.3466
- [5] WHO, «COVID-19 pandemic triggers 25% in-
crease in prevalence of anxiety and depression
worldwide,» 2 març 2022. [En línea]. Available:
[https://www.who.int/news/item/02-03-2022-covid-
19-pandemic-triggers-25-increase-in-prevalence-of-
anxiety-and-depression-worldwide](https://www.who.int/news/item/02-03-2022-covid-19-pandemic-triggers-25-increase-in-prevalence-of-anxiety-and-depression-worldwide).
- [6] N. B. Nuram Bayram, «The prevalence and socio-
demographic correlations, anxiety and stress among
a group of university students» 2008.
- [7] Magisnet, «Psicólogos evalúan a 14.000
alumnos en un pionero estudio sobre
salud mental,» [En línea]. Available:
[https://www.magisnet.com/2022/03/psicologos-
evaluan-a-14-000-alumnos-en-un-pionero-estudio-
sobre-salud-mental/](https://www.magisnet.com/2022/03/psicologos-evaluan-a-14-000-alumnos-en-un-pionero-estudio-sobre-salud-mental/)
- [8] La Vanguardia, «Más de mil llamadas
al teléfono del suicidio el día de su pu-
esta en marcha,» [En línea]. Available:
[https://www.lavanguardia.com/vida/20220511/
/8260274/telefono-prevencion-suicidio-024-
llamadas-primer-dia.html](https://www.lavanguardia.com/vida/20220511/8260274/telefono-prevencion-suicidio-024-llamadas-primer-dia.html).
- [9] Programming Dive, «Understanding MVC
design pattern,» [En línea]. Available:
[https://programmingdive.com/understanding-mvc-
design-pattern-in-php/](https://programmingdive.com/understanding-mvc-design-pattern-in-php/)
- [10] Spring, «Building REST services
with Spring,» [En línea]. Available:
<https://spring.io/guides/tutorials/rest/>
- [11] Spielberger, C.D., Gorsuch, R. y Lushene, R. (1970).
«Manual for the State-Trait Anxiety Inventory». Palo
Alto, CA: Consulting Psychologist.
- [12] Spielberger, C. D., Gorsuch, R. L. y Lushene, R.
E. (2008). «STAI. Cuestionario de ansiedad estado-
rasgo (7ª ed. rev.)». Madrid: TEA.
- [13] Beck, A. T., Ward, C. H., Mendelson, M., Mock, J., &
ERBAUGH, J. (1961). «An inventory for measuring
depression. Archives of general psychiatry,» 4(6),
561-571.
- [14] Cybermedian, «What is Model-View-Controller
(MVC) Framework? Model MVC with UML
Robustness Analysis,» [En línea]. Available:
[https://www.cybermedian.com/what-is-model-view-
controller-mvc-framework-model-mvc-with-uml-
robustness-analysis/](https://www.cybermedian.com/what-is-model-view-controller-mvc-framework-model-mvc-with-uml-robustness-analysis/).
- [15] Hamid Shaikh, «Sending Email using Spring
Boot Rest API,» [En línea]. Available:
[https://www.onlyxcodes.com/2021/02/sending-
email-spring-boot-rest-api.html](https://www.onlyxcodes.com/2021/02/sending-email-spring-boot-rest-api.html).

APÈNDIX

A.1 Diagrama de classes de la base de dades

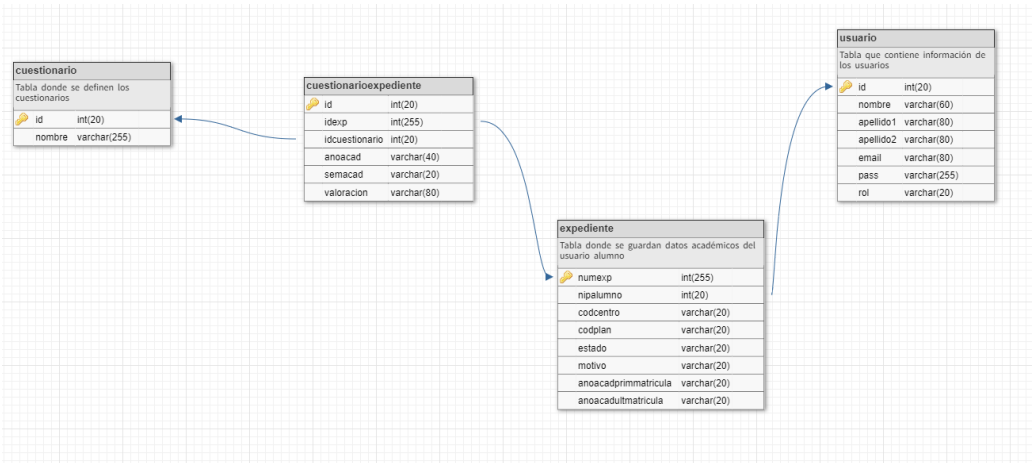


Fig. 12: Diagrama de classes de la base de dades del projecte

A.2 Diagrama de Gantt

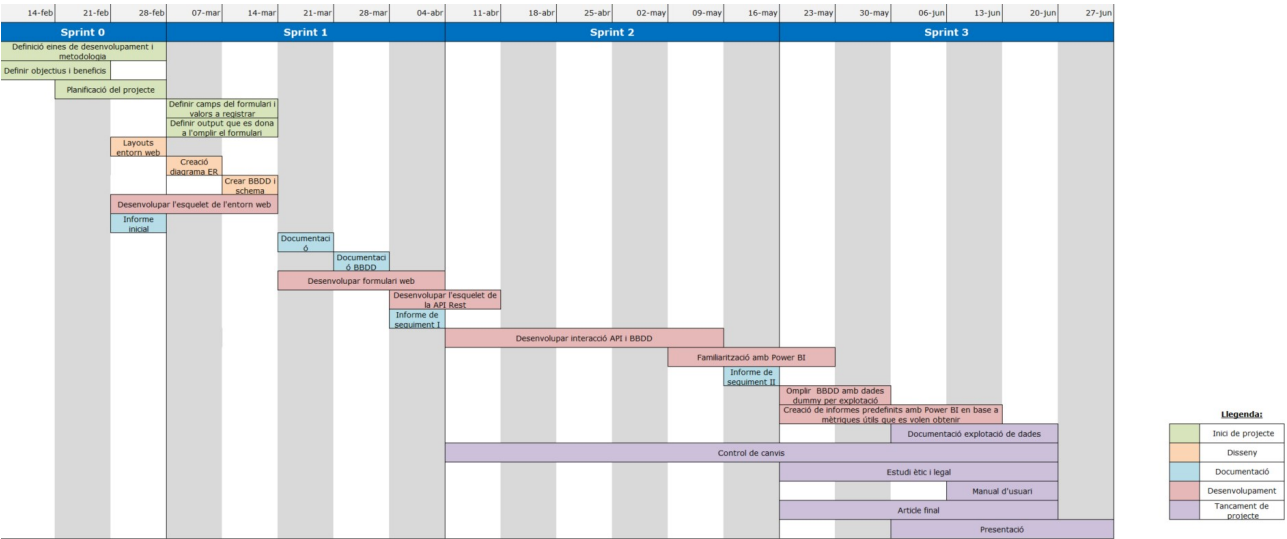


Fig. 13: Diagrama de Gantt del projecte

A.3 Codi de l'arxiu index.php

```

1 <?php session_start();
2
3 $action = $_GET['action'] ?? null;
4 switch ($action){
5     case '404':
6         require __DIR__.'404.php';
7         break;
8     case 'completarLogin':
9         require __DIR__.'resource_completarLogin
10 .php';
11         break;
12     case 'principal':
13         require __DIR__.'resource_principal.php';
14         break;
15     case 'nouQuestionari':
16         require __DIR__.'resource_nouQuestionari
17 .php';
18         break;
19     case 'logout':
20         require __DIR__.'resource_logout.php';
21         break;
22     case 'contestaQuestAR':
23         require __DIR__.'
24 resource_contestaQuestAR.php';
25         break;
26     case 'contestaQuestAE':
27         require __DIR__.'
28 resource_contestaQuestAE.php';
29         break;
30     case 'contestaQuestDE':
31         require __DIR__.'
32 resource_contestaQuestDE.php';
33         break;
34     case 'postAE':
35         require __DIR__.'resource_postAE.php';
36         break;
37     default:
38         require __DIR__.'resource_portada.php';
39         break;
40 }

```

Listing 1: Arxiu index.php de l'entorn web

A.4 Codi simplificat d'un arxiu resources

```

1 <?php require("resource_head.php");?>
2 <body>
3 <div class="d-flex" id="wrapper">
4     <?php require("resource_sidebar.php");?>
5     <!-- Page content wrapper-->
6     <div id="page-content-wrapper">
7         <?php require("resource_nav.php");?>
8         <div class="container-fluid">
9             <?php require __DIR__.'controlador/
10 principal.php'; ?>
11         </div>
12     </div>
13 </body>

```

Listing 2: Arxiu resource principal.php de l'entorn web

A.5 Codi simplificat de l'arxiu model

```

1 <?php
2
3 define('BASE_URL', 'http://localhost:8080/api/');
4
5 function callAPI($method, $url, $data = false)
6 {
7     $curl = curl_init();
8     switch ($method)

```

```

9 {
10     case "POST":
11         curl_setopt($curl, CURLOPT_POST, 1);
12         if ($data)
13             curl_setopt($curl,
14 CURLOPT_POSTFIELDS, $data);
15         break;
16     case "PUT":
17         curl_setopt($curl,
18 CURLOPT_CUSTOMREQUEST, "PUT");
19         if ($data)
20             curl_setopt($curl,
21 CURLOPT_POSTFIELDS, $data);
22         break;
23     case "DELETE":
24         curl_setopt($curl,
25 CURLOPT_CUSTOMREQUEST, "DELETE");
26         break;
27     default:
28         if ($data)
29             $url = sprintf("%s%s", $url,
30 http_build_query($data));
31 }
32 if ($data)
33 {
34     curl_setopt($curl, CURLOPT_HTTPHEADER,
35 array('Content-Type: application/json'));
36 }
37 curl_setopt($curl, CURLOPT_URL, $url);
38 curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1)
39 ;
40 $response = curl_exec($curl);
41 curl_close($curl);
42 return $response;
43 }
44
45 function addCuestionarioExpediente($data)
46 {
47     $url = BASE_URL . "cuestionarioexpediente/";
48     $method = "POST";
49     return callAPI($method, $url, $data);
50 }
51
52 function getQuest()
53 {
54     $url = BASE_URL . "cuestionarios/";
55     $method = "GET";
56     return callAPI($method, $url);
57 }

```

Listing 3: Fragment de l'arxiu conectaAPI.php de l'entorn web

A.6 Codi del controlador una vegada es contesta el qüestionari AE

```

1 <?php
2 if ($_SERVER["REQUEST_METHOD"] == "POST") {
3     $filters_respuestasNe = array (
4         'pregunta1' => FILTER_DEFAULT,
5         'pregunta2' => FILTER_DEFAULT,
6         'pregunta5' => FILTER_DEFAULT,
7         'pregunta8' => FILTER_DEFAULT,
8         'pregunta10' => FILTER_DEFAULT,
9         'pregunta11' => FILTER_DEFAULT,
10        'pregunta15' => FILTER_DEFAULT,
11        'pregunta16' => FILTER_DEFAULT,
12        'pregunta19' => FILTER_DEFAULT,
13        'pregunta20' => FILTER_DEFAULT
14    );
15    $filters_respuestasPos= array (
16        'pregunta3' => FILTER_DEFAULT,
17        'pregunta4' => FILTER_DEFAULT,
18        'pregunta6' => FILTER_DEFAULT,
19        'pregunta7' => FILTER_DEFAULT,

```

```

20     'pregunta9' => FILTER_DEFAULT,
21     'pregunta12' => FILTER_DEFAULT,
22     'pregunta13' => FILTER_DEFAULT,
23     'pregunta14' => FILTER_DEFAULT,
24     'pregunta17' => FILTER_DEFAULT,
25     'pregunta18' => FILTER_DEFAULT,
26 );
27 $filters_datos = array (
28     'idExp' => FILTER_DEFAULT,
29     'idCuestionario' => FILTER_DEFAULT,
30     'anoAcad' => FILTER_DEFAULT,
31     'semAcad' => FILTER_DEFAULT,
32 );
33
34 $data_filters_datos = filter_input_array(
35     INPUT_POST, $filters_datos);
36 $respuestas_positivas = filter_var_array(
37     $_POST['respuestas'], $filters_respuestasPos);
38 $respuestas_negativas = filter_var_array(
39     $_POST['respuestas'], $filters_respuestasNe);
40 $pos = array_sum($respuestas_positivas);
41 $neg = array_sum($respuestas_negativas);
42 $val = 30 + $pos - $neg;
43
44 if ($val > 18) {
45     if ($val > 19) {
46         if ($val > 28) {
47             $valoracion = "Alta";
48         } else $valoracion = "Sobre
49 Mitjana";
50     } else $valoracion = "Mitjana";
51 } else
52     $valoracion = "Tendencia Mitjana";
53
54 $data = array_merge($data_filters_datos,
55     array( 'valoracion' => $valoracion));
56 $data = json_encode($data);
57 addCuestionarioExpediente($data);
58
59 echo "<script>
60 var valoracion = '$valoracion';
61 redirigeEmail(2,valoracion)
62 </script>";
63 } else {
64     echo "<script> alert('Error');
65     window.location.href = './index.php?
66     action=404';
67     </script>";
68 }

```

Listing 4: Arxiu postAE.php de l'entorn web

A.7 Funció de JavaScript que passa dades al controlador per enviar email a l'alumne

```

1 function redirigeEmail (quest, valoracion) {
2     if (quest == 1) {
3         window.location.replace("./index.php?
4         action=enviaMailAR&valoracion=" + valoracion)
5     } if (quest == 2) {
6         window.location.replace("./index.php?
7         action=enviaMailAE&valoracion=" + valoracion)
8     } if (quest == 3) {
9         window.location.replace("./index.php?
10        action=enviaMailDE&valoracion=" + valoracion)
11    }
12 }

```

Listing 5: Funció redirigeEmail del fitxer functions.js

A.8 Codi que fa crida a la API per enviar un mail a l'alumne

```

1 <?php
2 $Val = $_GET['valoracion'];
3 $texto = null;
4
5 switch ($Val) {
6     case "Alta":
7         $texto = "";
8         break;
9     case "Sobre Mitjana":
10        $texto = "";
11        break;
12     case "Mitjana":
13        $texto = "";
14        break;
15     case "Tendencia Mitjana":
16        $texto = "";
17        break;
18     default:
19        echo "<script> alert('Error');
20        window.location.href = './index.php?
21        action=404';
22        </script>";
23 }
24 $Asunto = "Resultats";
25 $email = $_SESSION['email'];
26 $data = array_merge(array('to' => $email), array(
27     'mensaje' => $texto), array('asunto' =>
28     $Asunto));
29 $data = json_encode($data);
30 enviaEmail($data);

```

Listing 6: Versió resumida de l'arxiu enviaMailEA.php de l'entorn web

A.9 Codi simplificat d'un controlador de la API

```

1 @RestController
2 @RequestMapping("/api")
3 public class CuestionarioExpedienteController {
4
5     @Autowired
6     private CuestionarioExpedienteRepository
7         cuestionarioExpedienteRepository;
8
9     @PostMapping(value= "/cuestionarioexpediente",
10        consumes="application/json")
11     public CuestionarioExpediente
12     creaCuestionarioExpediente(@RequestBody
13     CuestionarioExpediente cuestionarioexpediente
14     ) {
15         return cuestionarioExpedienteRepository.save(
16             cuestionarioexpediente);
17     }
18
19     @ResponseBody
20     @GetMapping("/cuestionarioExpedientes/{idexp}")
21     public ResponseEntity<List<
22     CuestionarioExpediente>>
23     cuestionarioExpedientePorId(@PathVariable("
24     idexp") Long idexp) {
25
26         try {
27             List<CuestionarioExpediente>
28             cuestionarios = new ArrayList<
29             CuestionarioExpediente>();
30             cuestionarioExpedienteRepository.
31             findByIdexp(idexp)
32             .forEach(cuestionarios::add);
33         }
34     }
35 }

```

```

22         if (cuestionarios.isEmpty()) {
23             return new ResponseEntity<> (
24                 HttpStatus.NO_CONTENT);
25             }
26         return new ResponseEntity<> (
27             cuestionarios, HttpStatus.OK);
28         } catch (Exception e) {
29             return new ResponseEntity<> (null,
30                 HttpStatus.INTERNAL_SERVER_ERROR);
31         }
32     }

```

Listing 7: Versió resumida de l'arxiu CuestionarioExpedienteController.java de la API Rest

```

5     @Query(value = "SELECT *"
6         + " from cuestionarioexpediente"
7         + " WHERE idexp = ? AND semacad = ? AND
8         + " anoacad = ?"
9         + " GROUP BY idcuestionario, idexp",
10        nativeQuery = true)
11    List<CuestionarioExpediente>
12    findByIdexpAndSemacadAndAnoacad(Long idexp,
13        String semacad, String anoacad);

```

Listing 9: Arxiu CuestionarioExpedienteRepo.java de la API Rest

A.10 Codi simplificat d'un model de la API

```

1 @Entity
2 @Table(name = "cuestionarioexpediente")
3 @EntityListeners(AuditingEntityListener.class)
4 public class CuestionarioExpediente {
5
6     @Id
7     @GeneratedValue(strategy = GenerationType.
8         IDENTITY)
9     private Long id;
10    private String valoracion;
11    private Long idexp;
12    private Long idcuestionario;
13    private String anoacad;
14    private String semacad;
15
16    //Ara es defineixen les relacions entre
17    //taules
18    @ManyToOne
19    @JoinColumn(name = "idexp", insertable =
20        false, updatable = false)
21    private Expediente parentExpediente;
22
23    @ManyToOne
24    @JoinColumn(name = "idcuestionario",
25        insertable = false, updatable = false)
26    private Cuestionario parentCuestionario;
27
28    public CuestionarioExpediente() {}
29
30    public CuestionarioExpediente(Long id, String
31        valoracion, Long idExp, Long idCuestionario,
32        String anoAcad, String semAcad) {
33        this.id = id;
34        this.valoracion = valoracion;
35        this.idexp = idExp;
36        this.idcuestionario = idCuestionario;
37        this.anoacad = anoAcad;
38        this.semacad = semAcad;
39    }
40
41    //Aquí trobaríem els getters i setters i la
42    //funció toString per passar dades a aquest
43    //format
44    }

```

Listing 8: Versió resumida de l'arxiu CuestionarioExpedienteModelo.java de la API Rest

A.11 Codi simplificat d'un repositori de la API

```

1 @Repository
2 public interface CuestionarioExpedienteRepository
3     extends JpaRepository<CuestionarioExpediente,
4         Long>{
5
6     List<CuestionarioExpediente> findByIdexp(Long
7         idexp);
8
9 }

```

A.12 Codi simplificat del servei de la API per enviar emails

```

1 @Service
2 public class SolicitudEmailService {
3     public boolean sendEmail(String asunto, String
4         mensaje, String to)
5     {
6         boolean emailSent = false; // Variable que
7         // torna si un mail s'ha enviat correctament
8         String senderEmail = "stautfgtest@gmail.com";
9         // gmail email id del que fa enviament
10        String senderPassword = "hznzagelwvuipappf";
11
12        // Properties class que ens permet connectar
13        // amb el host del servidor SMTP
14        Properties properties = new Properties();
15        properties.put("mail.smtp.auth", "true");
16        // activa authentication
17        properties.put("mail.smtp.starttls.enable",
18            "true"); // activa TLS-protected connection
19        properties.put("mail.smtp.host", "smtp.
20            gmail.com"); // SMTP server de Gmail per
21            enviar email
22        properties.put("mail.smtp.port", "587"); //
23        // port TLS es el numero 587
24        Session session = Session.
25            getDefaultInstance(properties, new
26            Authenticator()
27            {
28                protected PasswordAuthentication
29                getPasswordAuthentication() {
30                    return new PasswordAuthentication(
31                        senderEmail, senderPassword);
32                }
33            });
34        try {
35            MimeMessage msg = new MimeMessage(
36                session);
37            msg.setFrom(new InternetAddress(
38                senderEmail));
39            msg.addRecipient(Message.RecipientType.
40                TO, new InternetAddress(to));
41            msg.setSubject(asunto);
42            msg.setText(mensaje);
43            Transport.send(msg); // Clase Transport
44            // per enviar el missatge
45            System.out.println("Mensaje enviado");
46            emailSent = true;
47        } catch (Exception e) {
48            System.out.println("EmailService
49            File Error"+ e);
50        }
51        return emailSent; // return emailSent
52        // variable
53    }
54 }

```

Listing 10: Versió de l'arxiu SolicitudEmailService.java de la API Rest