



UNIVERSITAT DE
BARCELONA

Treball final de grau
**GRAU D'ENGINYERIA
INFORMÀTICA**

Facultat de Matemàtiques
Universitat de Barcelona

**ReMemory: Sistema d'Avaluació
per al Tractament de Persones
amb Deteriorament Cognitiu Lleu**

Autor: Jaume Sánchez Castellanas

Director: Dra. Petia Radeva
Realitzat a: Departament de Matemàtica
Aplicada i Anàlisi. UB

Barcelona, 7 de setembre de 2016

Abstract

This project aims on developing a software to help with the diagnosis and tracking of people with mild cognitive impairment (MCI). The MCI is the loss of cognitive functions, especially memory deficits. This is a transitional state between normal aging and mild dementia and is usually an early sign of Alzheimer's disease. That is why it is important to diagnose it in time by temporary tracking of subjects. Currently there is no automated system to carry out the necessary tests to evaluate the progress of the disease. So this is the objective of the project: build an application capable of performing the relevant tests and save the results to analyze them later.

Resumen

Este proyecto se basa en crear un software que ayude a evaluar a personas con deterioro cognitivo leve (DCL). El DCL es la pérdida de funciones cognitivas, especialmente el déficit de memoria. Este constituye un estado transitorio entre el envejecimiento normal y la demencia leve y suele ser una señal precoz de la enfermedad de Alzheimer. Es por eso que es importante diagnosticarlo a tiempo haciendo un seguimiento temporal de los sujetos. Actualmente no hay ningún sistema automatizado que permita realizar los test necesarios para evaluar el progreso de la enfermedad. Es pues este el objetivo del proyecto; construir una aplicación capaz de efectuar las pruebas pertinentes, guardar sus resultados y así poderlos analizar posteriormente.

Resum

Aquest projecte es basa en crear un software que ajudi a avaluar a persones amb deteriorament cognitiu lleu (DCL). El DCL és la pèrdua de funcions cognitives, especialment el déficit de memòria. Aquest constitueix a un estat transitori entre l'enveliment normal i la demència lleu i sol ser una senyal precoç de la malaltia d'Alzheimer. És per això que és important diagnosticar-ho a temps fent un seguiment temporal dels subjectes. Actualment no hi ha cap sistema automatitzat que permeti realitzar els testos necessaris per avaluar el progrés de la malaltia. És doncs aquest l'objectiu del projecte; construir una aplicació capaç d'efectuar les proves pertinents, guardar els seus resultats i així poder-los analitzar posteriorment.

Agraïments

Vull agrair especialment al Gabriel de Oliveira per la seva constància, el seu suport i l'ajuda inestimable que m'ha proporcionat. També voldria expressar la meva gratitud a la doctora Petia Radeva per haver confiat en mi a l'hora de tirar endavant amb aquest projecte i a tot l'equip de ReMemory, que estan fent una gran labor ajudant a moltes famílies a reduir l'impacte que la malaltia suposa en les seves vides.

Espero humilment haver pogut contribuït, d'alguna manera, en la gran tasca humana que això suposa.

Keywords

Neuro-cognitive diseases, mild cognitive impairment, Alzheimer disease, memory impairment diagnosis, software aided treatments

Índex

1	Introducció	1
2	Treball Relacionat	3
3	Disseny	4
3.1	Plantejament	4
3.2	Requisits	4
3.2.1	Autentificació d'usuaris	4
3.2.2	Administració d'avaluadors	4
3.2.3	Administració de pacients	5
3.2.4	Tests	5
3.2.5	Guardar resultats	6
3.3	Estructura dels tests	6
3.3.1	Valoració cognitiva prèvia a la inclusió	6
3.3.2	Protocol neuropsicològic	7
3.4	Navegació	9
4	Implementació	11
4.1	Eines	11
4.2	Arquitectura	12
4.3	Gestió de dades	13
4.3.1	Gestió de dades	13
4.4	Tests i material mèdic	15
4.4.1	Fitxa del pacient	15
4.4.2	Valoració cognitiva prèvia	17
4.4.3	Sessió 1	19
4.4.4	Sessió 2	23
4.4.5	Valoració cuidador	24
4.5	Diagrama d'ús	26
4.6	Distribució	27
4.6.1	Empaquetat	27
4.6.2	Executable	27
4.6.3	Instal·lador	27

1 Introducció

Motivació

El deteriorament cognitiu lleu (DTL) és la pèrdua de memòria o disminució de funcions cognitives. El diagnòstic del DTL és complicat ja que sol ser un estat transitori entre l'envelliment normal i la demència, per tant s'han de realitzar proves eficaces per tal de separar els diferents casos.

En aquest projecte s'han implementat diferents paquets de tests per avaluar i entrenar els pacients. Alguns d'ells son realitzats més d'un cop en períodes de temps diferents per analitzar l'evolució dels subjectes. A part, per poder valorar la seva efectivitat, es compararan els resultats del desenvolupament d'aquests pacients amb altres que no han realitzat aquestes proves.

En aquest treball es busca implementar i automatitzar tests ja existents i comuns en diagnosis i tractament del DCL, Alzheimer o demència. Per tal de facilitar tot el procés de proves d'aquest sistema de tests, s'ha trobat la necessitat de crear una plataforma que els uneixi i que a més permeti fer un seguiment dels resultats dels pacients, podent-los analitzar i comparar.

Actualment no existeix cap sistema que disposi de les eines per dur a terme els tests de forma automàtica. Sí que existeix en el cas d'algun test, però el que més interessa és tenir-los tots junts i poder gestionar els pacients en un mateix programa.

És doncs aquest el principal repte del treball, crear un software que ajudi als metges a avaluar als pacients de forma senzilla i ordenada, guardant els resultats de tal manera que puguin ser analitzats o interpretats posteriorment.

Objectius

L'objectiu prioritari del treball és crear una eina que faciliti la feina dels evaluadors. Per fer això hi haurà comunicació constant amb ells per implementar totes les funcions que necessitin de la forma més clara possible.

Hi ha dos aspectes molt importants a tenir en compte: funcionalitat i disseny. El programa ha de funcionar i ha de ser intuïtiu. El que busquem doncs és donar facilitats, no complicacions. L'eina busca ser complementaria als mètodes manuals, tot buscant facilitar la gestió, evaluació i seguiment de pacients.

Per aconseguir-ho, durant el procés de creació s'aniran enviant versions del programa als evaluadors per tal de rebre retroavaluació i així poder corregir errors o afegir funcionalitats no contemplades anteriorment. En definitiva, per satisfer totes les seves necessitats.

És primordial que els evaluadors només hagin d'aplicar els seus coneixements a l'hora d'utilitzar el programa. L'aplicació ha de ser autocontinguda, i el cost d'aprenentatge d'ús d'aquesta hauria de ser nul o ínfim per tal de centrar-se en els interessos mèdics.

Organització del treball

Tasca	Inici	Final	Temps (dies)
Disseny	25/02/2016	28/02/2016	3
Gestió usuaris	28/02/2016	04/03/2016	5
Implementació Fitxa Pacient	05/03/2016	09/03/2016	4
Implementació Valoració Cognitiva Previa	10/03/2016	15/03/2016	5
Implementació 1ª Sessió Pacient	15/03/2016	08/04/2016	24
Implementació 2ª Sessió Pacient	09/04/2016	24/04/2016	15
Implementació Valoració Cuidador	25/04/2016	05/05/2016	10
Guardar resultats	06/05/2016	16/05/2016	10
Generar full de càcul	17/05/2016	25/05/2016	8
Crear versió portable	26/05/2016	01/06/2016	6
Revisió, Testeig i Debug	06/03/2016	30/06/2016	116
Reunions setmanals amb tutors	28/02/2016	08/07/2016	131
Escriure Memòria Projecte	25/06/2016	05/07/2016	10

Figura 1: Organització del treball

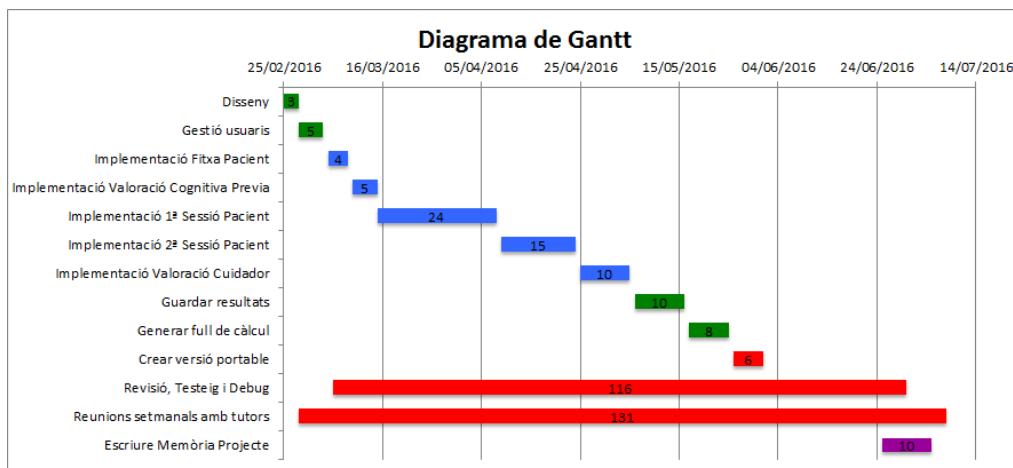


Figura 2: Diagrama de Gantt

2 Treball Relacionat

La peculiaritat d'aquest treball és el fet que el problema que volem afrontar és molt concret i per tant també necessitem una solució molt concreta.

És aquest doncs el motiu del treball, no hi ha cap eina que pugui satisfer les nostres necessitats i per això la creem.

Un dels propòsits del treball és automatitzar els tests que es realitzaran als pacients. Alguns d'aquests tests disposen de versions digitals automatitzades com en el cas del MoCA¹ [13], que estan desenvolupant una aplicació electrònica, o l'MMSE [6], que es pot trobar diverses webs on realitzar-lo.

Tot i existir aquests sistemes, en el nostre cas no ens acaben de ser útils ja que estem interessats en tenir ordenats i organitzats tots els tests en la mateixa aplicació, així com gestionar els diferents pacients per separat i aquestes requisits només els podem assolir implementant un software exclusiu.

¹www.mocatest.org

3 Disseny

3.1 Plantejament

El programa és exclusiu pels metges i les seves necessitats i per tant el disseny es basen en els requisits que ells demanen i els que deriven d'aquests.

Com que volem que l'aprenentatge d'ús de l'aplicació sigui molt ràpid, s'ha de procurar que la seva aparença sigui el màxim semblant al de les eines que ja s'han utilitzat fins ara, com són els testos en paper, la fulla de càlcul automatitzada o altres formularis. A més la interfície de navegació pels diferents menús ha de ser molt intuitiva i clara per facilitar-ne l'experiència d'usuari.

3.2 Requisits

Per implementar el programa necessitem els requisits que són donats primer pels avaluadors que hauran de treballar amb aquest sistema i a partir d'aquests requisits decidir quins altres fan falta.

3.2.1 Autentificació d'usuaris

Per tal d'aconseguir certa seguretat i que les dades dels pacients siguin accessibles només pels seus avaluadors, requerirem d'un sistema d'autentificació que valida si l'usuari està autoritzat per utilitzar el programa.

Hi ha dos tipus d'usuaris:

- Administrador
- Avaluadors

Els avaluadors poden accedir a les funcions per avaluar i gestionar els pacients, les quals en parlem en la secció 3.2.3. Hi ha un sol administrador i té la funció específica de gestionar els avaluadors.

3.2.2 Administració d'avaluadors

Aquest requisit només el pot realitzar un administrador havent accedit al sistema amb un nom d'usuari i contrasenya.

El que volem és que es pugui crear un nou avaluador que tingui accés a totes les funcions d'avaluació de pacients. Els avaluadors disposen de nom i de ID, així com d'una contrasenya per poder accedir al sistema. També s'han de poder editar posteriorment totes les dades sempre i quan el nom o la ID no coincideixin amb un altre avaluador. Finalment hi ha l'opció d'eliminar un avaluador.

3.2.3 Administració de pacients

Els pacients han de ser gestionats pels avaluadors. Aquests en poden crear, eliminar, importar o exportar. Un cop s'hagin afegit pacients, aquests poden ser seleccionats i es pot crear o editar la seva fitxa amb les dades necessàries pels metges.

A més també necessitem un sistema per exportar i importar pacients per facilitar la migració de dades entre ordinadors.

3.2.4 Tests

Implementem els tests de forma que s'assemblin al màxim als tests en paper intentant automatitzar-los tant com sigui possible, de forma que el cost d'aprenentatge per avaluadors que estiguin acostumat a tractar amb mètodes manuals sigui tan reduït com sigui possible.

Què vol dir que els tests siguin automàtics?

Hi ha alguns tests que el seu resultat es calcula mitjançant taules on es pondera el resultat directe segons l'edat i els anys d'escolarització dels pacients. La figura 3 correspon a la taula dels tests de díigits en el rang d'edats de 81-90 anys, d'aquesta taula en podem obtenir el “percentile range” i el “scaled score” que s'utilitza en la següent taula. Un cop tenim el “scaled score” de la taula de la figura 4 obtenim l'NSSA depenent de l'edat d'escolarització del pacient que està en un rang de 0-20 anys.

Scaled Score	Percentile Range	Digit Span	
		Forward	Backward
2	<1	≤3	0
3	1	—	2
4	2	—	—
5	3-5	4	—
6	6-10	—	—
7	11-18	—	—
8	19-28	—	—
9	29-40	5	3
10	41-59>	—	4
11	60-71	6	—
12	72-81	—	—
13	82-89	—	5
14	90-94	7	—
15	95-97	8	6
16	98	—	—
17	99	—	—
18	>99	9	7

Figura 3: Taula per calcular el percentil entre 50-56 anys

Els tests que inclou són els següents:

- Fulla de valoració clínica de demència (CDR)[2]
- Qüestionari de reserva cognitiva (CRC) [15]
- Test d'accentuació de paraules (TAP)[9]
- Mini-Mental State Examination (MMSE)[6]

3.3.2 Protocol neuropsicològic

Aquest protocol consta de tres sessions diferents que seran aplicades tres vegades (T1 - Basal, T2 - Fi de tractament, T3 - Seguiment 3 mesos), abans de la intervenció, després de la intervenció i després de 3 mesos de la finalització de la intervenció. La durada prevista del protocol és de 2 hores aproximadament.

A continuació expliquem les diferents sessions.

3.3.2.1 Primera sessió

En aquesta sessió es fan una sèrie de proves de memòria al pacient que seran guiades per l'avaluador.

Els tests d'aquesta sessió son els següents:

- Díigits ordre directe [20]
- Díigits ordre invers [20]
- Memòria lògica I [17]
- CogState ²
- Memòria lògica II Record lliure [17]
- Memòria lògica II Reconeixement [17]
- Boston Naming Test, short form version (BNT) [12]
- Color Trails [8]
- Five-digits test [18]
- Fluència verbal [4]

El CogState es realitza des d'un programa extern, per tant les dades d'aquest han de ser importades al nostre programa.

²<https://cogstate.com/>

3.3.2.2 Segona sessió

En aquesta sessió es fa a continuació de l'anterior i també la realitza el pacient supervisat per l'avaluador.

Els tests d'aquesta sessió son els següents:

- Cribatge: MoCA [13]
- UPSA versió reduïda (communication + comprehension/Planning domains) [10]
- Queixes subjectives de memòria (MFE) [5]
- Escala hospitalària d'ansietat i depressió (HAD)[21]
- QoL-AD [11]
- Duke Suport Social Index [3]
- Escala de Autoestima de Rosenberg (RSE) [16]

3.3.2.3 Valoració cuidador

Aquesta valoració a diferència de les dues anteriors la realitza el cuidador del pacient guiat per l'avaluador. És la última part del protocol.

Els tests d'aquesta sessió son els següents:

- FAQ (Functional Assessment Questionnaire) [14]
- NPI (Neuropersonal Psychiatric Assessment) [7]
- Depressió i Ansietat: Escala Hospitalaria de Ansiedad y Depresion (HAD) [21]
- Qüestionari Zarit[1]
- QoL-AD [11]
- Duke Suport Social Index [3]
- SF-12 [19]
- Duke Suport Social Index [3]

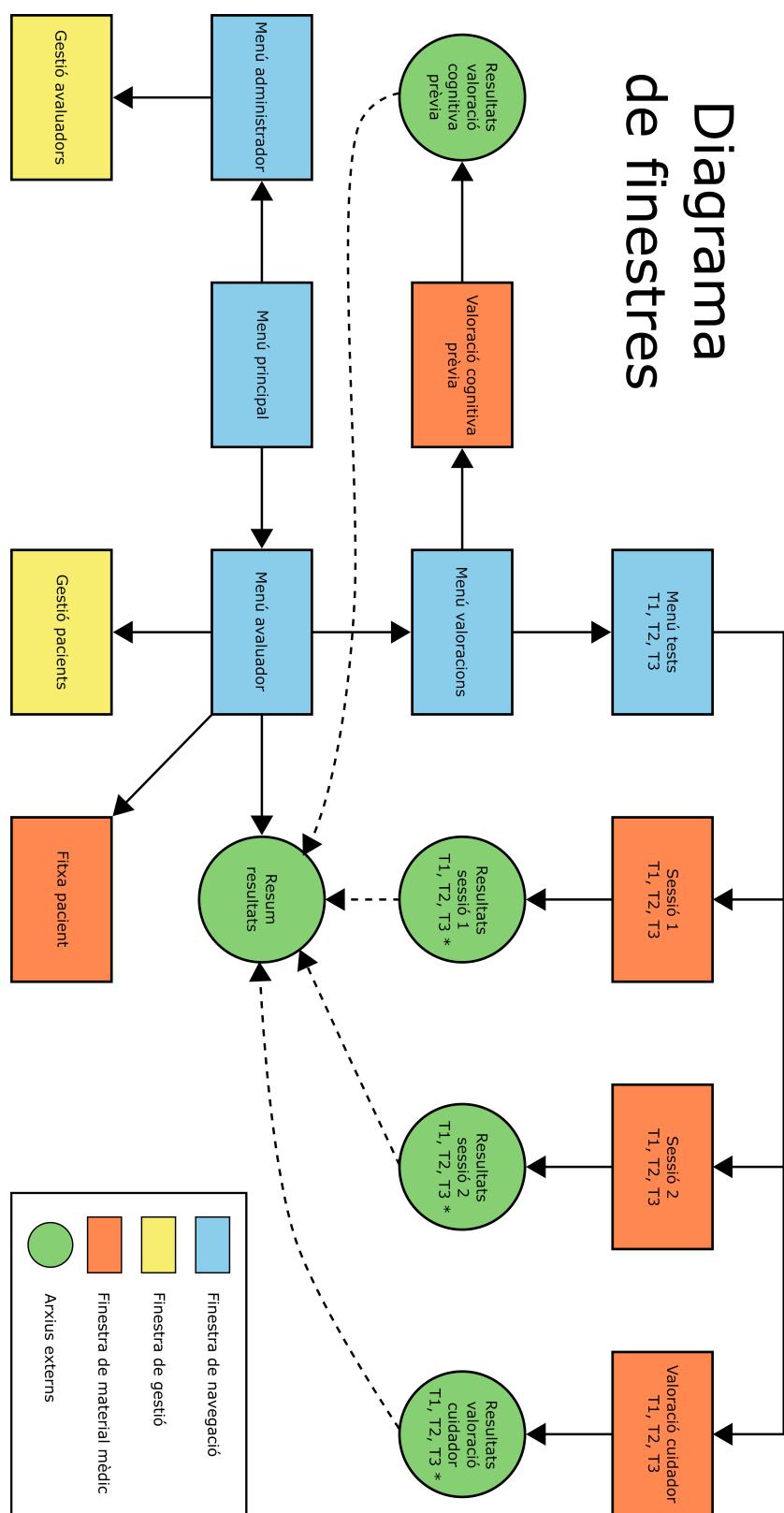
Com ja hem comentat, aquest protocol es realitza tres vegades. Per facilitar l'ús i per guardar les dades de cada protocol, els separem dins del programa en el menú de tests.

3.4 Navegació

Tenint els requisits hem de pensar quines finestres necessitem, quines funcions han de tenir i com està estructurat el programa. Per entendre com hem estructurat els requisits que hem definit anteriorment hem fet el diagrama de la figura 6.

Com es pot veure, del menú principal surten els menús de administrador i avaluator, dels quals ambdós tenen gestió d'usuaris, on els administradors gestionen evaluadors i els evaluadors gestionen als pacients. Del menú avaluator deriva el menú de valoracions que conté la valoració cognitiva prèvia i un nou menú per gestionar les tres diferents fases del protocol neuropsicològic. En el menú de tests es troben els tres apartats del protocol neuropsicològic i cada un d'ells guarda el resultat segons la seva fase (T1, T2, T3). Per tan entre aquests resultats i els de la valoració cognitiva prèvia hi ha un total de 10 arxius de resultats que s'utilitzen per crear un arxiu final de resultats total des del menú d'avaluator.

Diagrama de finestres



*Els resultats de les sessions 1 i 2 de la valoració cognitiva es guarden per separat per T1, T2 i T3 (hi ha 9 arxius de resultats individuals)

Figura 5: Diagrama conceptual de requisitos a finestres.

4 Implementació

4.1 Eines

Les eines amb les que hem treballat per construir el programa han estat seleccionades segons les necessitats i que hem creut millors per satisfer-les.

Per començar, a l'hospital treballen amb ordinadors amb Windows, per tan prescindirem de altres plataformes com iOS, Android o altres. Per raons de seguretat i privacitat, el programa no ha de treballar en xarxa ni necessita connexió a Internet, per aquest motiu, hem optat per fer una aplicació d'escriptori i no web.

Per programar el sistema hem decidit utilitzar Java, pels motius que hem comentat i per el domini del llenguatge adquirit durant la carrera ja que s'utilitza en moltes assignatures. A més, com que el programa en un futur pot ser utilitzat en altres centres, Java és multiplataforma i podria utilitzar-se en sistemes operatius com Linux o Mac OS.

Pel que fa a l'entorn de desenvolupament, hem utilitzat NetBeans IDE ja que té un editor de GUI molt complet, a més d'haver treballat molt amb ell durant la carrera. Un cop el projecte ha estat més avançat també hem utilitzat IntelliJ IDEA ja que és molt potent en navegació pel codi i en autocompletat de codi i això ajuda molt en projectes que comencen a tenir una certa envergadura.

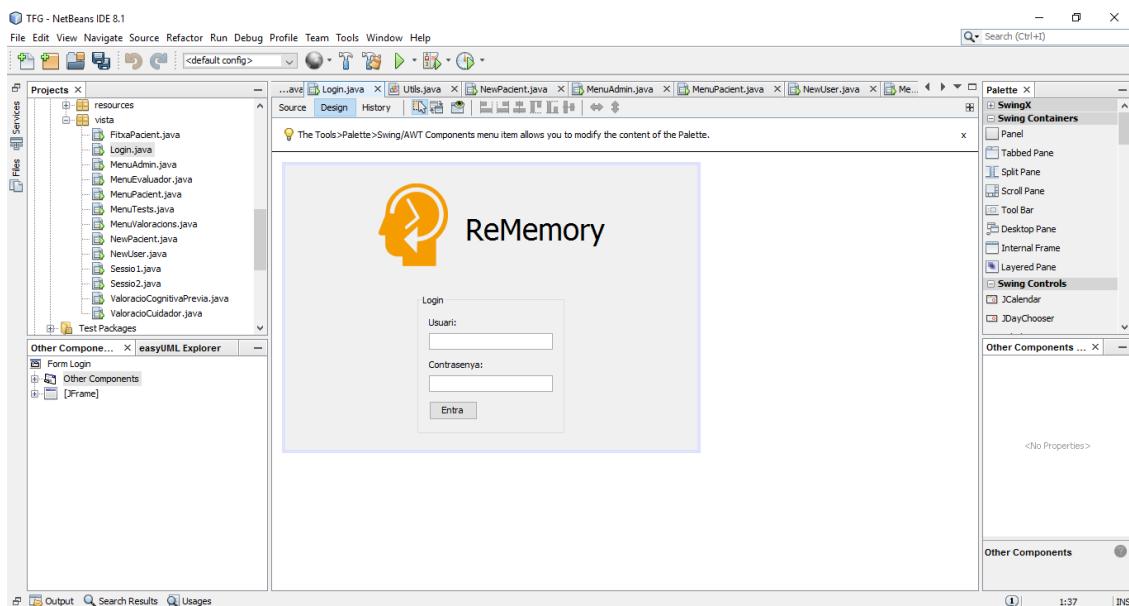


Figura 6: Disseny de finestres utilitzant NetBeans

Durant el procés de desenvolupament del programa ens hem trobat amb la necessitat de utilitzar llibreries externes per dur a terme algunes funcions específiques.

Les llibreries que hem utilitzat són les següents:

- org.json ³
- JCalendar ⁴
- SwingX ⁵
- Apache POI ⁶

4.2 Arquitectura

Per organitzar el codi hem utilitzat el patró model-vista-controlador (mvc) per facilitar la reutilització del codi i fer-lo més entenedor. Utilitzant aquest patró el diagrama de classes queda com es mostra a la figura 9.

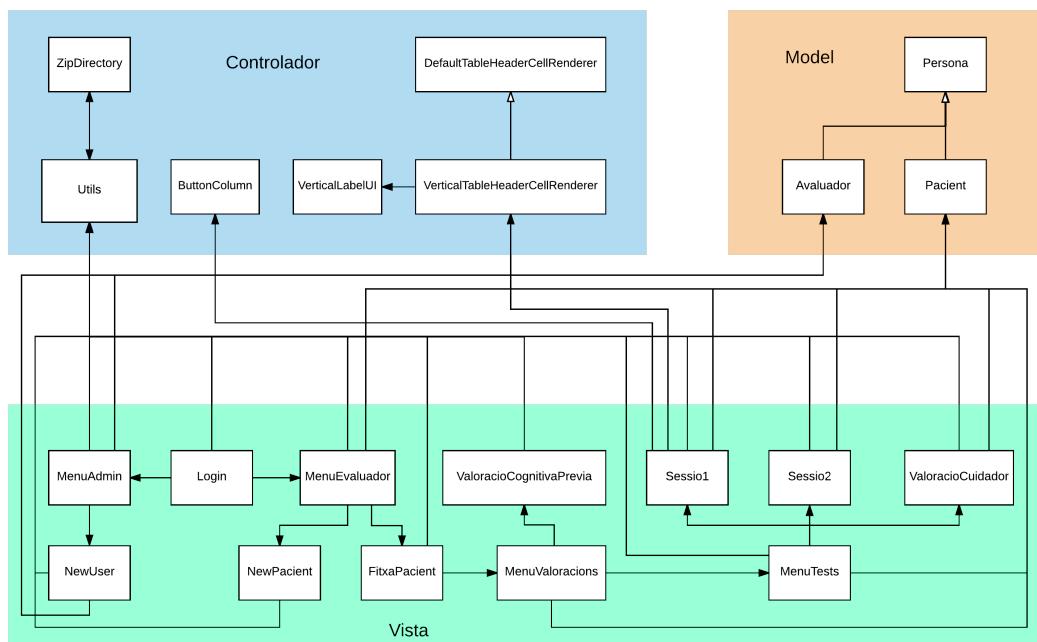


Figura 7: Diagrama de classes

Donada la complexitat dels tests i les seves funcions hem optat per implementar la lògica d'aquests a les classes del view en lloc d'extreure-ho al controlador, deixant només la part de lògica compartida entre classes de view a la classe Utils. La resta de classes del controlador són llibreries externes per facilitar algunes funcionalitats.

³<http://www.json.org/>

⁴<http://toedter.com/jcalendar/>

⁵<https://swingx.java.net/>

⁶<https://poi.apache.org>

4.3 Gestió de dades

Per la gestió de les dades del programa hem optat per prescindir de bases de dades ja que no disposem de grans quantitats de dades i hem utilitzat alternatives més senzilles.

Les dades del programa es guarden a la carpeta Roaming dins de AppData. Allà, dins d'una carpeta amb el nom del programa (ReMemory), hi haurà una carpeta de nom "res" (resources) amb l'arxiu de gestió d'usuaris i pdf utilitzats als testos i una carpeta per a cada pacient amb les seves dades.

4.3.1 Gestió de dades

Gestió d'usuaris

Per gestionar els usuaris hem utilitzat arxius json amb la informació necessària de cada un. Hem separat els usuaris amb rols: administrador, pacient i avaluador. Per llegir i escriure sobre aquests jsons, hem utilitzat la llibreria org.json. Aquestes dades serviran per la identificació d'usuaris dins del programa com pel login o per seleccionar un pacient. Com que el programa no fa servir internet i als ordinadors només hi han de tenir accés els metges, la seguretat de l'aplicació i de les dades dels pacients recau sobre la seguretat ja present a nivell hospitalari sobre l'accés dels metges als seus ordinadors de treball.

Resultats

Per guardar els resultats dels pacients hem utilitzat la classe Properties de Java que permet guardar les variables en un fitxer extern i després recuperar-les. Per cada bloc mèdic dels pacients com són la fitxa, la valoració cognitiva prèvia, la sessió 1, la sessió 2 i la valoració de cuidador, guardem un fitxer amb totes les dades dels seus camps per poder-les recuperar en sessions següents del programa. A part, en els que hem de guardar resultats (tots menys la fitxa del pacient), creem un altre fitxer amb els resultats que necessitarem per fer l'informe final del pacient. Cal tenir en compte que la sessió 1, la sessió 2 i la valoració de cuidador es realitzen tres vegades (T1, T2, T3) i per tant cada una d'elles tindrà tres fitxers de resultats i tres fitxers de camps.

La classe Properties de Java requereix d'una clau i un valor per establir les propietats que després seran recuperades. Per fer això hem creat un mètode per cada tipus de component que utilitzem en les finestres per tal de facilitar la programació. Així doncs a la classe Utils hem creat diferents mètodes "setProperty" que els hi passem la classe properties on s'ha de guardar, el nom de la property i el component que conté la dada que volem guardar. Fem això ja que cada component guarda diferent les dades, per exemple per obtenir el text d'un JTextField només cal cridar el mètode "getText()":

```
property.setProperty("nomPatient", jTextField1.getText());
```

En canvi amb un JComboBox hem d'obtenir l'item seleccionat i passar-lo a string "getSelectedItem().toString()".

```
property.setProperty("estatCivil",  
jComboBox1.getSelectedItem().getString());
```

Aquests només són dos exemples però treballem amb molts components diferents i amb aquests mètodes que hem creat, ens hem de despreocupar pel tipus de component i el resultat es simplifica pels anteriors exemples quedant així:

```
Utils.setProperties(properties, "nomPacient", jTextField1);  
Utils.setProperties(properties, "estatCivil", jComboBox1);
```

El mateix fem a l'hora de recuperar les dades creant mètodes "getProperties".

```
Utils.getProperties(properties, "nomPacient", jTextField1);  
Utils.getProperties(properties, "estatCivil", jComboBox1);
```

Les properties es carreguen a l'obrir la finestra corresponent, si es que s'ha guardat anteriorment, i es guarden quan es clica el botó accepta de la part inferior de la finestra.

Com que hi ha molts components i moltes dades a guardar hem creat un mètode a Utils que li passem el JPanel principal de la finestra que volem guardar i el nom del pacient i el test corresponents per tal de recuperar-ho posteriorment. El que fa aquest mètode és recórrer tots els components que hi ha dins el JPanel i guardar-los. Aquest és un mètode recurrent i quan troba un JPanel es torna a cridar a ell mateix per seguir buscant a dins d'aquest.

Aquest mètode s'utilitza per guardar les dades i els estats dels components com poden ser els JToggleButton o JCheckBox i les claus que utilitzem a les properties són els mateixos noms de les variables. Per recuperar les dades fem el mateix procés a la inversa, així doncs els noms de les claus no necessiten ser lògics en funció al seu contingut i ens estalviem nombrar molts components, ja que es trigaria molt a posar noms personalitzats a tots. Per guardar els resultats que necessitem per generar l'informe sí que posem noms personalitzats a les claus ja que haurem de seguir un ordre a l'hora de guardar-los, és per això que per aquests resultats creem un nou properties i així també separem conceptes.

Informe

Per poder analitzar els resultats dels pacients, els avaluadors necessiten les dades en format csv, ja que disposen d'un software que treballa en aquest format.

Per generar aquest informe utilitzem les dades dels resultats guardats per cada test i ho escrivim a un fitxer utilitzant la classe Writer de Java. Un avantatge del csv és que no cal cap mena de formatació especial, només cal separar els valors amb punt-i-coma, ja que el separador per defecte de csv a Espanya és el punt-i-coma, ja que utilitzem la coma pels decimals. Així doncs, el que fem és escriure les dades a un fitxer separant-les per punt-i-coma.

Una part dels resultats, els de CogState, venen donats per separat amb una fulla de càcul. Per gestionar-ho, a l'apartat de CogState de la sessió 1 l'usuari pot importar aquesta fulla de càcul que serà guardat a la carpeta de dades del pacient corresponent. Aquesta fulla de càcul serà utilitzat també per generar l'informe en csv. Per fer-ho utilitzem la llibreria Apache POI que permet llegir i escriure fitxers xls ixlsx. Així doncs es llegeixen les cel·les necessàries de la fulla de càcul i s'escriuen al csv de l'informe.

Migració de dades

És possible que es vulguin migrar dades d'un ordinador a un altre i per això hem implementat l'opció d'exportar i importar pacients.

Com hem dit les dades dels pacients es guarden en una carpeta per a cada un d'ells però com que el fitxer json dels usuaris està a la carpeta "res", necessitem extreure el pacient en un altre json i guardar-lo a la carpeta corresponent. Un cop fet això es comprimeix la carpeta del pacient i es posa de nom del fitxer resultant la id del pacient amb extensió "ReMemoryPatient" per remarcar que es tracta d'un arxiu del programa.

Per importar-lo es selecciona l'arxiu, es descomprimeix, es llegeix el json per recuperar les dades, es crea la carpeta del pacient, si aquesta no existeix, i es copia el contingut a dins. Si el pacient no existeix, aquest s'afegeix al json d'usuaris i si existeix només es copien les dades a la carpeta, sobreescritint les que hi pugui haver.

4.4 Tests i material mèdic

En aquest apartat explicarem com hem implementat la part de tests i material mèdic del programa. Repassarem tots els apartats explicant les peculiaritats de cada un, les eines que hem utilitzat i com hem afrontat el seu disseny.

La majoria de material ja té el seu disseny en format paper i hem hagut de decidir quins components fer servir i com estructurar-los per tal de reproduir el disseny original i així facilitar la feina als avaluadors.

Tots els materials tenen diferents apartats. Per separar-los de forma visual utilitzem JTabledPanes i així queden dividits en diferents "tabs"

4.4.1 Fitxa del pacient

Per la fitxa del pacient els avaluadors disposen d'una fulla de càcul que té diferents apartats separats per fulles diferents. Com que els apartats els separen amb tabs, el resultat visual és similar al de Microsoft Excel amb les fulles.

Dades bàsiques

En aquesta fulla hi ha dades que l'input és text i altres que hi ha diferents valors definits per seleccionar. Per fer aquest disseny hem optat per utilitzar JTextFields i JTextField per cada un d'aquests inputs.

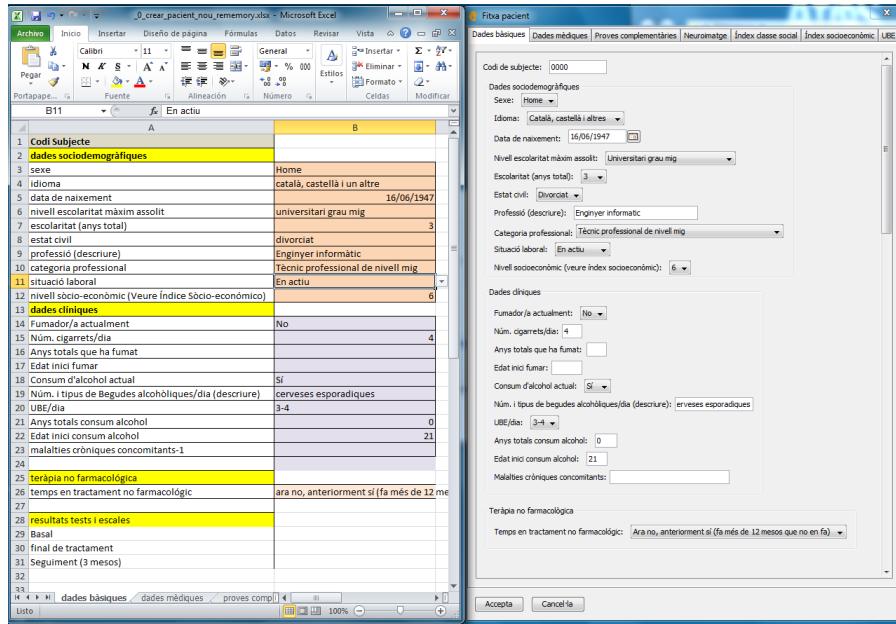


Figura 8: Comparació fitxa de pacient

Dades mèdiques

A les dades mèdiques es demana una dates en els antecedents del pacient. Per gestionar la introducció de dates en el programa hem utilitzat la llibreria JCalendar que afegeix un component swing que permet seleccionar la data amb un calendari.

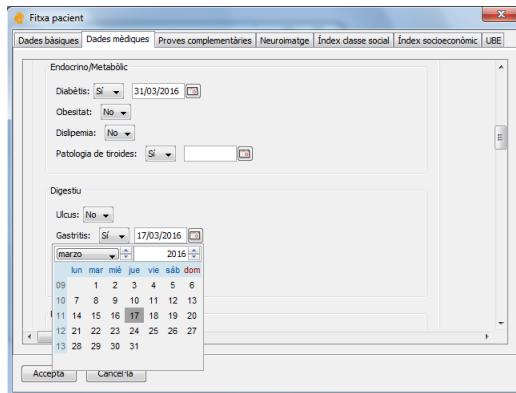


Figura 9: Exemple de JCalendar

Proves complementaries i neuroimatge

Aquests dos apartats els hem dissenyat amb els mateixos components que els anteriors ja que no necessiten cap altre recurs especial.

Índex classe social, índex socioeconòmic i UBE

Aquest apartat conté imatges amb taules informatives, per evitar complicacions hem afegit aquestes imatges directament al programa.

4.4.2 Valoració cognitiva prèvia

La valoració cognitiva prèvia conté 4 apartats: CDR[2], CRC[15], MMSE[6] i TAP[9].

CDR

El CDR té 10 pàgines diferents, per separar-les al programa hem utilitzat un CardLayout i així dins del JPanel del "tab" del CDR hi ha un JPanel per a cada pàgina i només se'n mostrerà una. Per gestionar la pàgina que es mostra utilitzem dos botons, "anterior" "següent", que permeten navegar entre les pàgines. Aquest sistema és utilitzat en altres apartats que necessiten diverses pàgines.

En aquest apartat utilitzem JTextField per inputs de text curts, JTextArea per texts llargs, JComboBox per llistes de resultats i JCheckBox per opcions que s'han de marcar amb una creu.

Al final del CDR hi ha una taula per calcular la puntuació. Per tal de calcular la puntuació de forma automàtica hem copiat la taula utilitzant JTogglesButtons per seleccionar els elements de la taula, sumem les puntuacions individuals de cada selecció i fem la mitjana aritmètica. Amb la mitjana definim el tipus de deteriorament.

VALORACION CLINICA DE DEMENCIA (CDR)					
	0	0.5	1	2	3
Memoria	Ninguna 0	Dudosa 0.5	Leve 1	Moderada 2	Grave 3
	Sin pérdida de memoria o leves olvidos inconstantes; recobro parcial de eventos; olvidos temporales	Pérdida moderada de memoria; más marcada que la memoria de trabajo; defecto interfiere con las actividades diarias	Pérdida grave de memoria; sólo resiente información recién aprendida; materiales nuevas se pierden con rapidez	Pérdida grave de memoria; sólo resiente fragmentos	
Orientación	Completamente orientado pero con leve dificultad para recordar fechas y temporales	Dificultad moderada con las relaciones temporales; orientación alterada; desorientado en el tiempo; a menudo en el lugar	Dificultad grave con las relaciones temporales; orientación muy alterada; desorientado en el tiempo, a menudo en el lugar	Orientado sólo en persona	
Razonamiento y solución de problemas	Resuelve los problemas diarios y se encarga bien de las finanzas; razonamiento burocrático; comportamiento previo	Leve dificultad para resolver problemas; similitudes y diferencias	Dificultad moderada para hacer frente a problemas; razonamiento social limitado	Hacer grandes dificultades para hacer frente a problemas; razonamiento social limitado	Incapaz de razonar o resolver problemas
Actividades fuera de casa	Función independiente a su nivel habitual en el trabajo, como voluntariado y agrupaciones sociales	Leve dificultad en estas actividades	Incapaz de ser independiente fuera de casa	Parece estar lo suficiente bien como para realizar funciones fuera de casa	Parece demasiado débil para realizar funciones fuera de casa
Actividades domésticas y aficiones	Vida en casa, aficiones e intereses intelectuales bien conservados	Vida en casa, aficiones e intereses intelectuales algo limitados	Dificultad leve pero clara del razonamiento; abandono de las tareas más difíciles; algunas aficiones y intereses más complicados	Sólo realiza tareas simples; intereses muy limitados	Sin función significativa en casa
Cuidado personal	Completamente capaz de cuidarse por sí misma	Necesita recordatorios	Requiere ayuda para vestirse, asearse y encargarse de sus efectos personales	Requiere mucha ayuda para su cuidado personal; incontinencia frecuente	

VALORACIÓ CLÍNICA DE DEMÈNCIA (CDR)					
	Deteriorament				
	Cap 0	Dubtós 0.5	Leu 1	Moderat 2	Greu 3
Memòria	Sense pèrdua de memòria o les oblidis inconstants	Oblidis leus constants; recol·lecció parcial d'estendiments; defecte interferix amb les relacions temporals	Pèrdua moderada de memòria; més oblidis cada dia; defecte interferix amb els aprenentatges	Pèrdua greu de memòria; nombrosos aprenentatges amb molt interferix; matèries noves es perdren amb rapidesa	Pèrdua greu de memòria; només resten fragments
Orientació	Completement orientat	Completement orientat però amb lleu dificultat per acomplir tarefas	Completement orientat però amb lleu dificultat per acomplir tarefas	Dificultat moderada amb les relacions temporals; orientació alterada; pot tenir una idea del lloc habitual desorientat en el temps; sovint en el lloc desorientat geogràfica en el temps	Dificultat greu amb les relacions temporals; orientació molt alterada; pot tenir una idea del lloc habitual desorientat en el temps; sovint en el lloc desorientat geogràfica en el temps
Razonament i solució de problemes	Resol els problemes diaris i se encarga bien de las finanzas; razonamiento burocrático; comportamiento previo	Leu dificultat per acomplir problemes; similituds i diferencies; razonamiento social habitual mantenido	Leu dificultat per acomplir problemes; similituds i diferencies	Dificultat moderada per acomplir problemes; similituds i diferencies; razonamiento social habitual mantingut	Greus dificultats per fer front a problemes; similituds i diferencies; judio social desorientat
Activitats fora de casa	Incapacitat per ser independent en estas actividades	Incapacitat per ser independent en estas actividades	Incapacitat per aquestes activitats	Incapacitat per aquestes activitats	Incapacitat per acomplir tarefas
Activitats domèstiques i aficions	Necessita recordatori	Requiere ayuda para vestirse, asearse y encargarse de sus efectos personales	Necesita recordatori	Necesita recordatori	Sense fundició significativa a casa
Cuidat personal	Completament capaz de cuidar-se per si mateixa	Requiere mucha ayuda para vestirse, asearse y encargarse de sus efectos personales	Requiere ayuda per vestirse, asearse y encarregar-se dels seus efectes personals	Requiere molta ajuda per al seu cuidat personal; incompletes freqüents	

Puntú només quan dominava el seu以前 habitual a causa de pèrdua cognitiva, no a limitacions degudes a altres factors.

Valoració clínica de demència (CDR): 1,4 (deteriorament leu)

Figura 10: Comparativa CDR original amb CDR ReMemory

CRC

En el CRC hem decidit utilitzar JRadioButtons ja que s'ha de seleccionar una opció entre diverses i ens interessa poder-les veure totes. Per això no hem utilitzat JComboBox ja que aquests s'han d'expandir per veure les opcions.

Cada opció seleccionada del CRC suma punts. Amb el total d'aquests punts s'obté el rang segons uns intervals. Aquest rang es calcula automàticament amb el programa i es mostra a la part inferior.

- $Punt \leq 6$ = quartil 1 ($\leq C1$), rang Inferior
- Punt 7 y 9 = quartil 1-2 ($C1-C2$), rang mig-baix
- Punt 10 y 14 = quartil 2-3 ($C2-C3$), rang mig-alt
- $Punt \geq 15$ = quartil 4 ($\geq C4$), rang superior.

MMSE

En l'MMSE hi han opcions per marcar amb una creu, per tant hem utilitzat JCheckBox al disseny. Cada JCheckBox seleccionat suma 1 a la puntuació total i a partir d'aquesta puntuació obtenim la puntuació de referència segons a quin interval estigui.

- ≥ 27 : normal
- 24-27: sospita patològica
- 12-24: deteriorament
- ≤ 12 : demència

TAP

En el TAP es seleccionen les frases que s'han accentuat bé. Per tal d'aconseguir el coeficient intel·lectual a partir dels resultats disposem de la taula de la figura 11

D'aquesta taula n'hem pogut extreure una fórmula per tal de fer més fàcil la implementació al codi:

$$CI = 2(total - 1) + 60 - modificador$$

on

$$modificador = 0 \text{ si } total < 5,$$

$$modificador = 1 \text{ si } total > 4 \text{ i } total \leq 20,$$

$$modificador = 2 \text{ si } total > 20$$

Conversion of TAP score to estimated full scale IQ (values in the right hand columns are FSIQs predicted at each level of TAP performance, as calculated using the regression model in the healthy subjects).

TAP score	Predicted FSIQ	TAP score	Predicted FSIQ
1	60	16	89
2	62	17	91
3	64	18	93
4	66	19	95
5	67	20	97
6	69	21	98
7	71	22	100
8	73	23	102
9	75	24	104
10	77	25	106
11	79	26	108
12	81	27	110
13	83	28	112
14	85	29	114
15	87	30	116

Figura 11: Taula coeficient intel·lectual TAP

4.4.3 Sessió 1

Els apartats de la Sessió 1 són: Dígits directe i invers [20], MLI i MLII [17], CogState⁷, BNT [12], color trails [8], five digit [18], fluència verval [4].

Automatització de taules

En aquesta sessió molts tests disposen de les taules esmentades a l'apartat de disseny, com les de les figures 3 i 4. En el cas de la taula de percentil, el resultat depèn de l'edat del pacient i de la puntuació obtinguda. A més retorna dos resultats, el percentil i la qualificació escalada utilitzada per obtenir el NSSA. Per aconseguir automatitzar això hem optat per utilitzar un HashMap de HashMaps així:

```
private HashMap<String , HashMap<Integer , List<Object>>>
taulaDigitsDD ;
```

Per a cada rang d'edats afegim un HashMap posant la puntuació com a clau i els dos resultats de valor. Aquests dos resultats els ajuntem utilitzant una llista d'objectes, ja que el percentil és un String i el NSSA és un enter.

Així doncs l'automatització de la taula de dígits directe pel rang d'edats de 50 a 56 queda així:

```
taulaDigitsDD = new HashMap<>();
List<Object> auxList ;
HashMap<Integer , List<Object>> auxMap = new HashMap<>();

auxList = Arrays.asList(2 , "<1");
auxMap.put(0 , auxList);
auxMap.put(1 , auxList);
auxMap.put(2 , auxList);
auxMap.put(3 , auxList);
auxList = Arrays.asList(5 , "3-5");
```

⁷<https://cogstate.com/>

```

auxMap.put(4, auxList);
auxList = Arrays.asList(9, "29-40");
auxMap.put(5, auxList);
auxList = Arrays.asList(11, "60-71");
auxMap.put(6, auxList);
auxList = Arrays.asList(14, "90-94");
auxMap.put(7, auxList);
auxList = Arrays.asList(15, "95-97");
auxMap.put(8, auxList);
auxList = Arrays.asList(18, ">99");
auxMap.put(9, auxList);
taulaDigitsDD.put("50-56", new HashMap(auxMap));

```

En aquest cas, el test de dígits, de taules com aquesta n'hi ha 10 per diferents rangs d'edat.

Per calcular el NSSA, veiem que la relació de la qualificació escalada amb la puntuació NSSA és lineal segons intervals d'edat d'escolarització. En l'exemple de la figura 4 aquests intervals són entre 0-2, 3-7, 8-12, 13-16, 17-20. En el primer interval (0-2) la diferència entre les puntuacions és de 2, i en cada interval que segueix aquesta diferència disminueix un punt. Així doncs la implementació d'aquest exemple queda així:

```

if (anysEscola <= 2){
    modificador = 2;
}
else if (anysEscola >= 3 && anysEscola <= 7){
    modificador = 1;
}
else if (anysEscola >= 8 && anysEscola <= 12){
    modificador = 0;
}
else if (anysEscola >= 13 && anysEscola <= 16){
    modificador = -1;
}
else if (anysEscola >= 17){
    modificador = -2;
}
nssaDD.setText(scaledScore+modificador+"");

```

Marcar dates

Als avaluadors els interessa saber en quin moment han fet els tests. Per fer això en els tests que es calcula el resultat amb un botó, com els dígits i la Memòria lògica (ML1 i ML2), un cop es calcula el resultat queda marcada la data a la part inferior del resultat utilitzant un label. En els que no hi ha aquest botó hem afegit un botó per marcar aquesta data.

Implementació del disseny

Pel fer el disseny d'aquests tests ens hem basat en el disseny que ja estava fet en la fulla de càlcul. Per tal d'aconseguir un resultat similar al disseny existent hem decidit utilitzar JXTable de la llibreria SwingX de Java. Aquesta llibreria ofereix més opcions que la llibreria Swing que ofereix per defecte l'entorn Java. Una de les opcions que hem utilitzat és la de pintar les files de les taules i així hem aconseguit el resultat que veiem a la figura 12 que podem comparar amb el disseny anterior a la figura 13

Digits directe				
Ítem	Intent	Resposta	Puntuació intent (0-1)	Puntuació ítem
1	9 - 7	9 - 7	1	
	6 - 3	6 - 3	1	2
2	5 - 8 - 2		0	
	6 - 9 - 4	6 - 9 - 4	1	1
3	7 - 2 - 8 - 6		0	
	6 - 4 - 3 - 9	6 - 4 - 3 - 9	1	1
4	4 - 2 - 7 - 3 - 1	4 - 2 - 7 - 3 - 1	1	
	7 - 5 - 8 - 3 - 6		0	1
5	3 - 9 - 2 - 4 - 8 - 7	3 - 9 - 2 - 4 - 8 - 7	1	
	6 - 1 - 9 - 7 - 4 - 2		0	1
6	4 - 1 - 7 - 9 - 3 - 8 - 6		0	
	6 - 9 - 1 - 7 - 4 - 2 - 8		0	0
7	3 - 8 - 2 - 9 - 6 - 1 - 7 - 4		0	
	5 - 8 - 1 - 3 - 2 - 6 - 4 - 7		0	0
8	2 - 7 - 5 - 8 - 6 - 3 - 1 - 9 - 4		0	
	7 - 1 - 3 - 9 - 4 - 2 - 5 - 6 - 8		0	0

SpanDD
 (Max = 9) Puntuació directa
 orden directo (DD)
 (Max = 16)

Calcular Percentil range:
 NSSA NEURONORMA Scaled Score age-adjusted

Reset Data del test: 05/07/2016 12:25:06

Figura 12: Disseny igits directe ReMemory

Orden directo				
Ítem	Intento	Respuesta	Puntuación Intento (0 ó 1)	Puntuación ítem (0, 1 ó 2)
1	9 - 7		0	0
	6 - 3		0	
2	5 - 8 - 2		0	0
	6 - 9 - 4		0	
3	7 - 2 - 8 - 6		0	0
	6 - 4 - 3 - 9		0	
4	4 - 2 - 7 - 3 - 1		0	0
	7 - 5 - 8 - 3 - 6		0	
5	3 - 9 - 2 - 4 - 8 - 7		0	0
	6 - 1 - 9 - 7 - 4 - 2		0	
6	4 - 1 - 7 - 9 - 3 - 8 - 6		0	0
	6 - 9 - 1 - 7 - 4 - 2 - 8		0	
7	3 - 8 - 2 - 9 - 6 - 1 - 7 - 4		0	0
	5 - 8 - 1 - 3 - 2 - 6 - 4 - 7		0	
8	2 - 7 - 5 - 8 - 6 - 3 - 1 - 9 - 4		0	0
	7 - 1 - 3 - 9 - 4 - 2 - 5 - 6 - 8		0	

SpanDD
 (Max = 9) Puntuación directa
 orden directo (DD)
 (Max = 16)

Reset Calcular Percentil Range
 NSSA NEURONORMA Scaled Score age-adjusted

Figura 13: Digits directe de la fulla de càlcul

Per completar el disseny de les taules hem utilitzat dos codis externs. Un d'ells és ButtonColumn que permet inserir fàcilment botons a les columnes de les taules, ja que per defecte és complex. L'altre codi és VerticalTableHeaderCellRenderer, que hem utilitzat per poder posar títols de columnes en vertical, ja que la llibreria JXTable no ho permet.

Per poder decidir quines cel·les es poden editar i quines no, hem creat un DefaultTableModel personalitzat que hem utilitzat en les taules. Dins d'aquesta classe hem afegit una matriu de booleans que s'editen utilitzant els mètodes "setEditable" que utilitzem en el mètode abstracte isCellEditable que retorna si la cel·la és editable o no.

Per definir colors de les capçaleres hem creat un DefaultTableCellRenderer personalitzat que li passem per paràmetre el color que volem.

Per pintar les files alternant colors hem utilitzat el mètode "setHighlighters", propi de la llibreria JXTable.

Digits, ML1 i ML2

Aquests tests tenen taules pels resultats, per tant hem utilitzat les tècniques definides anteriorment per automatitzar-los.

En els tests de dígits, els resultats de les files s'aparellen de dos en dos. Aquest aparellament serveix per calcular l'Span, per sumar un punt a l'Span el pacient ha de realitzar amb èxit una de les dos proves d'aquest bloc. El test s'acaba quan el pacient falla dos proves seguides o quan arriba al final.

BNT

El resultat del BNT també es calcula amb taules fent servir el sistema explícit.

Aquest test consta de tres parts i cada una d'elles es realitza en cada valoració diferent del protocol (T1, T2, T3). Així doncs quan es carrega la finestra de sessió 1 es comprova de quina valoració es tracta i es mostra el test BNT corresponent.

Color trails, five digit, fluència verbal

El resultat d'aquests tests també es calcula amb taules però aquestes taules són molt extenses i no hem pogut implementar la automatització per manca de temps. Per solventar-ho permetem a l'avaluador entrar les dades manualment i hem afegit un botó per mostrar les taules en format pdf.

En els tests color trail i five digit hem implementat cronòmetres ja que s'ha de calcular temps i amb aquesta eina facilitem la feina als avaluadors. Per fer-ho hem utilitzat timers i DecimalFormat per donar format.

4.4.4 Sessió 2

Els apartats de la Sessió 2 són: MoCA [13], UPSA [10], MFE [5], HAD [21], QoL-AD [11], DUKE [3], RSE [16].

Alguns d'aquests tenen qüestionaris on cada pregunta té x respuestes amb una puntuació diferent cada una. Per seleccionar les respuestes hem utilitzat JToggel-Buttons. Podem veure el resultat comparant les figures 14 i 15.

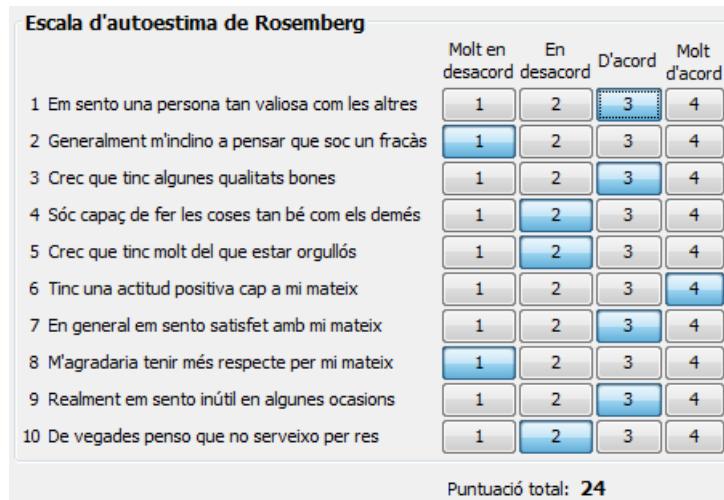


Figura 14: Disseny de RSE ReMemory

		Muy en desacuerdo	En desacuerdo	De acuerdo	Muy de acuerdo
1	Me siento una persona tan valiosa como las otras	1	2	3	4
2	Generalmente me inclino a pensar que soy un fracaso	1	2	3	4
3	Creo que tengo algunas cualidades buenas	1	2	3	4
4	Soy capaz de hacer las cosas tan bien como los demás	1	2	3	4
5	Creo que no tengo mucho de lo que estar orgulloso	1	2	3	4
6	Tengo una actitud positiva hacia mí mismo	1	2	3	4
7	En general me siento satisfecho conmigo mismo	1	2	3	4
8	Me gustaría tener más respeto por mí mismo	1	2	3	4
9	Realmente me siento inútil en algunas ocasiones	1	2	3	4
10	A veces pienso que no sirvo para nada	1	2	3	4

Total: _____

Figura 15: RSE versió paper

Per actualitzar la puntuació segons els botons que s'han seleccionat, hem creat un mètode a la classe Utils que afegeix un listener a cada botó que crida un altre mètode quan es aquest es clica. Aquest altre mètode recorre tots els botons i suma les seves puntuacions.

MoCA

Pel disseny de MoCA hem utilitzat JCheckBox per seleccionar les respostes correctes. A més hem afegit imatges del test original que són necessàries per realitzar el test.

UPSA

En el test UPSA hem utilitzat JTogglesButtons per poder seleccionar les respostes ja que aquestes estan marcades.

MFE, QoL-AD, DUKE, RSE

Aquests tests tenen els qüestionaris que hem explicat al principi d'aquesta secció per tant els hem implementat amb el sistema que hem comentat.

HAD

El disseny del HAD l'hem implementat com el CRC (secció 4.4.2) però amb diferents intervals.

Com CRC però alternant parells i senars

4.4.5 Valoració cuidador

Els apartats de la valoració del cuidador són: NPI [7], HAD [21], ZARIT [1], SF-12 [19], DUKE [3], FAQ [14].

NPI

Pel disseny del NPI hem utilitzat també JTogglesButtons per la freqüència i la gravetat i el resultat s'obté multiplicant les seleccions d'aquests dos. Hem inclòs també un JCheckBox a cada tipus de trastorn per decidir si el resultat és o no valorable, tal i com està definit al test original. Si aquest JCheckBox es selecciona el resultat d'aquesta valoració no suma a la puntuació total.

HAD, DUKE

Aquests tests són iguals que els de la sessió 2.

ZARIT, FAQ

Els apartats ZARIT I FAQ també inclouen qüestionaris com els de la sessió 2.

SF-12

En el test SF-12 hem utilitzat JRadioButtons per seleccionar les respostes.

4.5 Diagrama d'ús

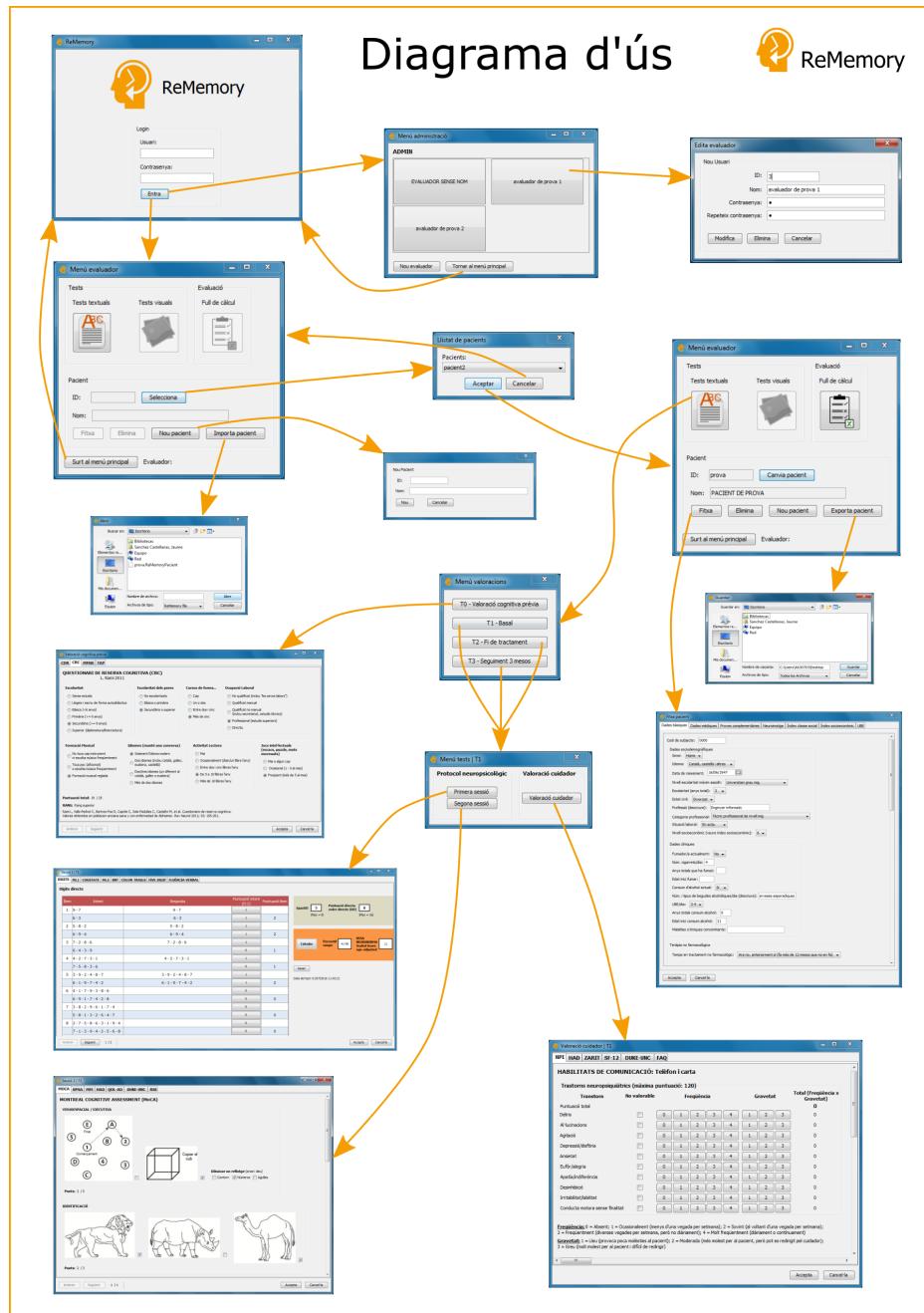


Figura 16: Diagrama d'ús de l'aplicació ReMemory

4.6 Distribució

Per distribuir el programa hem decidit crear un executable per Windows, ja que és el sistema operatiu que utilitzen al centre mèdic col·laborador. A més també hem creat un instal·lador per facilitar la inclusió del programa en els equips.

Per evitar dependències de Java hem inclòs la versió de Java jre que necessitem pel programa dins del paquet que instal·larem.

4.6.1 Empaquetat

Per poder crear l'executable del programa satisfactòriament hem inclòs totes les llibreries dins del mateix paquet .jar de l'aplicació. Per fer-ho hem afegit un target al build.xml que ajunta les llibreries i l'executem. Com a resultat obtenim un .jar amb tot el contingut que utilitzarem per crear l'executable.

4.6.2 Executable

Els programes Java per ser executats necessiten un conjunt d'utilitats instal·lades a l'equip, el JRE. En el aquest es disposa de la JVM (màquina virtual de Java) i el conjunt de llibreries i classes bàsiques de Java així com diversos arxius de suport. Per evitar dependències de Java en els equips on instal·larem el programa, hem inclòs el JRE dins el paquet d'instal·lació i l'hem enllaçat amb l'executable. Per tal de crear l'executable i enllaçar-lo hem utilitzat el programa Launch4j⁸ que és gratuït i ens permet fer això.

4.6.3 Instal·lador

Com que les dades del programa han d'estar a la carpeta Roaming de AppData i volem facilitar la feina als avaluadors, hem creat un instal·lador que ubica la carpeta de dades on li pertoca i deixa seleccionar on instal·lar l'executable.

Per crear l'instal·lador hem utilitzat el programa InnoSetup⁹ que ens permet personalitzar-lo amb diferents paràmetres.

⁸<http://launch4j.sourceforge.net/>

⁹<http://www.jrsoftware.org/>

5 Conclusions

El deteriorament cognitiu lleu (DCL) afecta a un 20% de la població major de 60 anys i la meitat d'aquests desenvolupa Alzheimer en un termini d'aproximadament cinc anys més. Tota la gent implicada amb el projecte ReMemory està fent una gran feina per alleujar les conseqüències que implica aquesta malaltia als afectats, tant els pacients com els familiars.

En aquest projecte hem creat una eina per a facilitar al personal sanitari la seva labor de diagnòstic i tractament de persones amb DCL. Gràcies al programa, hem complert el nostre propòsit: gestionar els pacients, les seves dades i que automatitzar al màxim les tasques de realització dels tests necessaris pel tractament de la malaltia. Tot això fent ús d'una interfície intuïtiva utilitzant elements similars a les eines ja conegeudes, creant un entorn familiar pels avaluadors.

El disseny dels tests com els digits [20] utilitzant JXTable per la semblaça amb les fulles de càcul o del MoCA [13] adjuntant les imatges del test original, han sigut un èxit segons les primeres reaccions que hem obtingut dels avaluadors.

L'automatització dels tests digits [20], memòria lògica [17] i BNT [12] és un punt a valorar molt favorablement dins aquest projecte. Gràcies a aquesta funcionalitat disminueix molt el temps d'avaluació dels tests permetent als avaluadors centrar-se exclusivament en els afers mèdics.

El fet de disposar de les dades dels pacients en el mateix programa, així com la possibilitat d'importar-les i exportar-les permetent la migració, permet una fàcil gestió i tractament d'aquestes.

Arrel d'aquest programa s'ha dotat al personal sanitari d'una eina fins ara inexistent que permet un millor tractament i estudi d'un seguit de malalties neurocognitives, proporcionant una millor atenció als pacients, amb la conseqüent millora en les seves qualitats de vida.

Treball Futur

Aquest projecte no s'acaba aquí, hi ha moltes millores a fer en diferents àmbits i es procurarà que aquestes s'apliquin, donant encara més potencial al programa.

Una part del projecte ReMemory es basa en crear tests visuals a partir de imatges que s'han recollit usant una càmera que els pacients portaran penjada com un collaret. D'aquestes imatges se'n seleccionen les més interessants usant algoritmes de visió per computador i amb aquestes s'elaboraran tests de memòria pels pacients. Quan aquests tests siguin una realitat, s'afegiran al programa dins la opció "tests visuals" que ara està desactivada.

Una altra tasca important a realitzar és automatitzar els tests que no s'han pogut implementar. És una labor costosa i es necessita de temps per poder-la realitzar.

Finalment hi ha el propòsit d'implementar un sistema de creació de blocs de tests personalitzats. Per això s'haurà de separar tots els tests existents en classes

separades i per tal de poder-los afegir en una finestra nova personalitzada. Això a més, ens servirà per tenir el codi molt més estructurat.

Referències

- [1] Michel Bédard, D William Molloy, Larry Squire, Sacha Dubois, Judith A Lever, and Martin O'Donnell. The zarit burden interview a new short version and screening version. *The gerontologist*, 41(5):652–657, 2001.
- [2] Leonard Berg. Clinical dementia rating (cdr). *Psychopharmacology bulletin*, 24(4):637–639, 1987.
- [3] WE Broadhead, Stephen H Gehlbach, Frank V De Gruy, and Berton H Kaplan. The duke-unc functional social support questionnaire: Measurement of social support in family medicine patients. *Medical care*, pages 709–723, 1988.
- [4] C Carnero-Pardo and A Lendínez-González. Utilidad del test de fluencia verbal semántica en el diagnóstico de demencia. *Revista de neurología*, 29(8):709–714, 1999.
- [5] Pedro Montejo Carrasco, Mercedes Montenegro Peña, and Manuel J Sueiro. The memory failures of everyday (mfe) test: Normative data in adults. *The Spanish journal of psychology*, 15(03):1424–1431, 2012.
- [6] JR Cockrell and MF Folstein. Mini-mental state examination (mmse). *Psychopharmacology bulletin*, 24(4):689, 1988.
- [7] Jeffrey L Cummings. The neuropsychiatric inventory assessing psychopathology in dementia patients. *Neurology*, 48(5 Suppl 6):10S–16S, 1997.
- [8] Lou D'Elia and Paul Satz. *Color trails test*. Psychological Assessment Resources, 1996.
- [9] Jesús J Gomar, Jordi Ortiz-Gil, Peter J McKenna, Raymond Salvador, Bibiana Sans-Sansa, Salvador Sarró, Amalia Guerrero, and Edith Pomarol-Clotet. Validation of the word accentuation test (tap) as a means of estimating premorbid iq in spanish speakers. *Schizophrenia research*, 128(1):175–176, 2011.
- [10] R Walter Heinrichs, Marta Statucka, Joel Goldberg, and Stephanie McDermid Vaz. The university of california performance skills assessment (upsa) in schizophrenia. *Schizophrenia research*, 88(1):135–141, 2006.
- [11] Kellee Howard and Kenneth Rockwood. Quality of life in alzheimer's disease. *Dementia and Geriatric Cognitive Disorders*, 6(2):113–116, 1995.
- [12] Wendy J Mack, David M Freed, Betsy White Williams, and Victor W Henderson. Boston naming test: shortened versions for use in alzheimer's disease. *Journal of gerontology*, 47(3):P154–P158, 1992.
- [13] Ziad S Nasreddine, Natalie A Phillips, Valérie Bédirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey L Cummings, and Howard Chertkow. The montreal cognitive assessment, moca: a brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4):695–699, 2005.

- [14] RI Pfeffer, TT Kurosaki, CH Harrah, JM Chance, and S Filos. Measurement of functional activities in older adults in the community. *Journal of gerontology*, 37(3):323–329, 1982.
- [15] Lorena Rami, Cinta Valls-Pedret, David Bartrés-Faz, Claudia Caprile, Cristina Solé-Padullés, Magdalena Castellví, and JL Molinuevo. Cuestionario de reserva cognitiva. valores obtenidos en población anciana sana y con enfermedad de alzheimer. *Revista de Neurología*, 52(4):195–201, 2011.
- [16] Morris Rosenberg. Rosenberg self-esteem scale (rse). *Acceptance and commitment therapy. Measures package*, 61, 1965.
- [17] Wechsler D Wechsler Memory Scale—Fourth. Edition (wms-iv). *New York: Psychological Corporation*, 2009.
- [18] M Sedó. Test de los cinco dígitos: five digit test. *TEA Ediciones, Madrid*, 2005.
- [19] John E Ware, Susan D Keller, and Mark Kosinski. *Sf-12: How to Score the Sf-12 Physical and Mental Health Summary Scales*. QualityMetric Incorporated, 1998.
- [20] David Wechsler. *WAIS-III: escala de inteligencia Wechsler para adultos-III*. El Manual Moderno, 1997.
- [21] Anthony S Zigmond and R Philip Snaith. The hospital anxiety and depression scale. *Acta psychiatrica scandinavica*, 67(6):361–370, 1983.