

PM

NexusBRaNT
Backoffice para o BRaNT
PROJETO DE MESTRADO

Érica Filipa Dias Cunha
MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

novembro | 2022

NexusBRaNT

Backoffice para o BRaNT

PROJETO DE MESTRADO

Érica Filipa Dias Cunha

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTAÇÃO

Filipe Magno de Gouveia Quintal

COORIENTAÇÃO

Yuri Aristides da Silva Godinho de Almeida



Faculdade de Ciências Exatas e da Engenharia

Mestrado em Engenharia Informática

2021/2022

NexusBRaNT - *Backoffice* para o BRaNT

Tese de Mestrado

Érica Filipa Dias Cunha, nº 2033617

Orientador: Filipe Magno Gouveia Quintal

Coorientador: Yuri Aristides da Silva Godinho de Almeida

novembro de 2022

Resumo

Na área da saúde, a utilização de tecnologias está em crescimento e pode ser aplicadas, em áreas como: gestão de recursos médicos, de pacientes, financeira e clínica. A utilização de sistemas para gestão de pacientes e reabilitação cognitiva está em constante evolução e atualmente existem algumas plataformas nesta área, cada qual com as suas vantagens, mas também com diversas limitações. Esta tese está enquadrada no projeto de investigação *Belief Revision applied to Neurorehabilitation Therapy* (BRaNT - Projeto PTDC/CCI-COM/30990/2017 financiado pela FCT).

Foi desenvolvida uma plataforma *online*, chamada NexusBRaNT, sendo esta o *backoffice* do projeto BRaNT. Esta plataforma *web* tem como público-alvo os profissionais de saúde que realizam reabilitação cognitiva (psicólogos, neuropsicólogos, terapeutas, entre outros). Tem como principais funcionalidades a gestão de pacientes, a gestão de avaliações neuropsicológicas e a gestão de treinos cognitivos.

Paralelamente à plataforma desenvolvida, foi implementada uma API REST que dá suporte à plataforma *web* e ao sistema externo das atividades cognitivas. Tem como principal objetivo responder aos pedidos solicitados, sendo estes: inserção, atualização, eliminação e obtenção dos dados pretendidos para serem apresentados na plataforma ou solicitados por outros elementos do projeto para as atividades cognitivas desenvolvidas pelos mesmos.

Foram realizados testes de usabilidade a 16 participantes, divididos em 3 grupos: estudantes de mestrado, profissionais de saúde da área de psicologia e as profissionais de saúde do projeto BRaNT. O protocolo do teste de usabilidade incluiu algumas fases, tais como: a utilização do sistema e a resposta do questionário SUS. Através dos testes de usabilidade foi possível realizar avaliação qualitativa e quantitativa acerca da plataforma. Ambas as avaliações tiveram resultados positivos, sendo a variedade instrumentos de avaliação à escolha, os dados normativos, a interpretação clínica nas avaliações neuropsicológicas, o cálculo automático do perfil cognitivo e da interpretação clínica e a visualização da performance e progresso nas sessões de treinos, as funcionalidades mais enaltecidas pelos participantes. De modo geral os participantes indicaram que a plataforma era intuitiva, de fácil utilização, perceptível e apelativa. Além disso, o sistema implementado foi também utilizado durante um mês e meio num estudo piloto desenvolvido pelo projeto de investigação, onde pode-se comprovar o bom funcionamento da plataforma *web* e da API (tanto no NexusBRaNT como no sistema externo).

Palavras-chaves – *healthcare information systems*, profissionais de saúde, reabilitação cognitiva remota, plataforma *web*, *frontend*, *backend*

Abstract

In healthcare, the use of technology is growing and can be applied in areas such as: medical resource management, patient management, financial management, and clinical management. The use of systems for patient management and cognitive rehabilitation is constantly evolving and there are currently a number of platforms in this area, each with its advantages, but also with several limitations. This thesis is framed within the research project Belief Revision applied to Neurorehabilitation Therapy (BRaNT - Project PTDC/CCI-COM/30990/2017 funded by FCT).

An online platform was developed, called NexusBRaNT, to access the backoffice of the BRaNT project. This web platform targets health professionals who perform cognitive rehabilitation (psychologists, neuropsychologists, therapists, among others). Its main features are patient management, neuropsychological assessment management, and cognitive training management.

In parallel to the developed platform, a REST API was implemented that supports the web platform and the external system of cognitive activities. Its main objective is to respond to the requested requests, which are: inserting, updating, deleting, and obtaining data intended to be presented in the platform or requested by other elements of the project for the cognitive activities developed by them.

Usability tests were performed to 16 participants, divided into 3 groups: master's students, health professionals from the psychology area, and the health professionals from the BRaNT project. The usability test protocol included some phases, such as: using the system and answering the SUS questionnaire. Through the usability tests it was possible to perform qualitative and quantitative evaluations about the platform. Both evaluations had positive results, with the variety of evaluation instruments to choose from, the normative data, the clinical interpretation in neuropsychological evaluations, the automatic calculation of the cognitive profile and clinical interpretation, and the visualization of performance and progress in training sessions being the functionalities most praised by the participants. In general, participants indicated that the platform was intuitive, easy to use, understandable, and appealing. In addition, the implemented system was also used for a month and a half in a pilot study developed by the research project, where the good functioning of the web platform and API (both in NexusBRaNT and in the external system) could be verified. Keywords - healthcare information systems, healthcare professionals, remote cognitive rehabilitation, web platform, frontend, backend.

Keywords - healthcare information systems, healthcare professionals, remote cognitive rehabilitation, web platform, frontend, backend

Agradecimentos

Agradeço ao meu orientador, Dr. Filipe Quintal, pela disponibilidade ao longo do projeto, e pela maneira objetiva de esclarecer as dúvidas durante o todo processo.

Agradeço ao meu coorientador, professor Yuri Almeida, por toda a disponibilidade, esclarecimento de dúvidas e colaboração nesta tese.

Agradeço a ambos os orientadores também pela orientação prestada e terem confiado em mim para realização deste projeto.

Agradeço a todos os elementos do projeto BRaNT que me acolheram bem, dispostos a esclarecerem dúvidas e a colaborarem com esta tese de mestrado. Um agradecimento especial à Dra. Ana Lúcia, à psicóloga Joana Câmara, à professora Teresa Paulino e ao professor Yuri que participaram constantemente e colaboraram arduamente para realização desta tese.

Agradeço meu namorado, André, por todo o apoio dado durante este processo e nunca ter me deixado desistir, mesmo quando não conseguia ver soluções em obstáculos que foram surgindo.

Agradeço à Micaela Jardim, por todo apoio dado e motivação, que mesmo longe estava sempre presente e quis acompanhar todo o processo. Agradeço aos meus colegas, que se tornaram amigos, Daniel Barros, Jocelyne Pestana, Guilherme Freitas, Luana Mendonça, Jénifer Constantino, Raquel Barros, Lisandro Marote e Carolina Sousa, com quem compartilhei os bons e os maus momentos ao longo da minha vida académica.

Agradeço ao meu pai, à Conceição e ao meu irmão pelo apoio incondicional ao longo destes cinco anos e pela motivação diária em não desistir. Ao meu pai que sempre se esforçou e se dedicou à família.

Finalmente agradeço a todos os meus colegas que acompanharam ao longo de 5 anos, aqueles que com as suas críticas e ideias ajudaram a melhorar este trabalho.

À minha mãe Ana Maria, a minha saudade eterna

Índice

Resumo	i
Abstract.....	iii
Agradecimentos	v
Índice	vii
Índice de Figuras	xi
Índice de figuras nos anexos	xii
Índice de Tabelas	xiv
Índice de tabelas nos anexos	xiv
Índice de Gráficos.....	xvi
Lista de Acrónimos.....	xvii
1. Introdução.....	1
1.1. Contexto.....	2
1.2. Problema	4
1.3. Solução proposta.....	4
1.4. Estrutura do relatório	6
2. Revisão de literatura.....	7
2.1. Défice cognitivo ligeiro (DCL).....	7
2.2. Sistemas existentes	7
2.2.1. Sistemas de reabilitação cognitiva.....	8
2.2.1.1. COGWEB.....	8
2.2.1.2. CogniFit.....	9
2.2.1.3. NeuronUp.....	10
2.2.1.4. RehaCom.....	11
2.2.2. Noutras áreas	12
2.2.3. Pontos fortes e fracos do sistema existente	13
2.2.4. Análise comparativa dos sistemas estudados	14
2.3. Ferramentas tecnológicas.....	15
2.3.1. Backend	15
2.3.2. Frontend.....	16
2.3.3. Base de dados	17
2.3.4. Ferramentas seleccionadas	17
3. Especificação do problema.....	19
3.1. Requisitos.....	19

3.1.1.	Evolução dos requisitos	20
3.1.2.	Requisitos de tecnologia.....	21
3.1.3.	Requisitos funcionais.....	21
3.1.3.1.	Requisitos funcionais gerais do sistema.....	21
3.1.3.2.	Requisitos funcionais da gestão de pacientes.....	22
3.1.3.3.	Requisitos funcionais da avaliação neuropsicológica	23
3.1.3.4.	Requisitos funcionais do programa treino.....	23
3.1.4.	Requisitos não funcionais.....	24
3.1.4.1.	Requisitos não funcionais de interface.....	24
3.1.4.2.	Requisitos não funcionais de <i>software</i> e <i>hardware</i>	24
3.1.4.3.	Requisitos não funcionais de sistemas complementares	24
3.1.4.4.	Requisitos não funcionais de interoperabilidade.....	25
3.1.5.	Casos de utilização	25
3.2.	Arquitetura proposta	27
3.3.	Método de desenvolvimento	28
4.	Implementação	29
4.1.	Ferramentas de desenvolvimento.....	29
4.2.	Base de dados.....	29
4.2.1.	Visão geral.....	31
4.2.2.	Dificuldades.....	32
4.3.	<i>Backend</i>	33
4.3.1.	Configurações.....	34
4.3.1.1.	Ligação à base de dados	34
4.3.1.2.	Servidor	35
4.3.1.3.	Definição de constantes e variáveis de ambiente.....	36
4.3.2.	Visão geral.....	37
4.3.3.	Rotas	38
4.3.4.	Controladores	43
4.3.5.	Modelos	45
4.3.6.	<i>Middlewares</i> e validadores	47
4.4.	<i>Frontend</i>	48
4.4.1.	Visão geral.....	49
4.4.2.	Comunicação frontend/backend	51
4.4.3.	Design.....	52
4.4.4.	Componentes customizáveis.....	53

4.4.5.	Funcionalidades	57
4.4.5.1.	Utilizadores comuns (PS clínico/investigador).....	58
4.4.5.2.	Administrador.....	68
4.5.	Conclusões	72
5.	Avaliação.....	73
5.1.	Procedimento dos testes de usabilidade.....	73
5.2.	Participantes.....	75
5.3.	Avaliação ao sistema	76
5.3.1.	Avaliação qualitativa	76
5.3.1.1.	Estudantes de mestrado	76
5.3.1.2.	Profissionais de saúde	77
5.3.1.3.	<i>Experts</i> do BRaNT	78
5.3.2.	Avaliação quantitativa	79
5.4.	Conclusões	84
6.	Discussão.....	87
6.1.	NexusBRaNT.....	87
6.2.	API.....	88
6.3.	Contribuições para a área.....	89
7.	Conclusão	91
7.1.	Trabalho futuro	92
8.	Referências	93
Anexos	97
Anexo A	– Informações base de dados.....	97
Anexo A.1	– Base de dados antiga	106
Anexo A.2	– Base de dados atual	107
Anexo B	– Funcionalidades	128
Anexo C	– Avaliação do sistema.....	135

Índice de Figuras

Fig. 1.1 – Arquitetura do projeto BRaNT: Patient Profiling – criação do perfil do paciente que integra várias avaliações neuropsicológicas para determina o estado cognitivo base; Training Selection – são definidos as tarefas de treino cognitivo mais apropriadas para a terapia neuropsicológica; Training Session – descrição das sessões de treino realizadas pelo paciente em casa; System Calibration – descrição da calibração que o sistema realiza comparando o perfil previsto com os dados reais do ficheiro PS recém-adquiridos.....	3
Fig. 2.1 – Exemplos dos gráficos de progresso na plataforma COGWEB [26].....	9
Fig. 2.2 – Alguns exemplos de exercícios realizados pelos pacientes na plataforma COGWEB [21].	9
Fig. 2.3 – Interface do CogniFit: a) programação da sessão de treino; b) gestão do paciente e visualização gráfica dos resultados [22].	10
Fig. 2.4 – Interface do NeuronUp: a) gestão de atividades; b) gestão de resultados [23].	11
Fig. 2.5 – Interface do RehaCom - módulo atenção e concentração: a) resultados; b) treino cognitivo [24].	12
Fig. 3.1 – Processo geral dos requisitos.	19
Fig. 3.2 – Diagrama de Casos de Utilização: Funcionamento Geral do NexusBRaNT.	26
Fig. 3.3 – Arquitetura proposta do NexusBRaNT com 4 camadas: presentation layer (interação entre o utilizador e o sistema), logic layer (manipulação da informação e solicitação da informação na camada inferior), data layer (armazenamento dos dados) e common layer (autorizações em comum nas camadas).	27
Fig. 4.1 – Organização dos ficheiros na API.....	33
Fig. 4.2 – Função “query”.	35
Fig. 4.3 – Função "transaction".	35
Fig. 4.4 – Definição inicial do endereço das rotas.....	36
Fig. 4.5 – Funcionamento geral da API.....	37
Fig. 4.6 – Funcionamento da rota de registo da informação clínica do paciente.	37
Fig. 4.7 – Organização das rotas na API.	38
Fig. 4.8 – Organização dos controladores na API.	43
Fig. 4.9 – Função “checkValidation”.	44
Fig. 4.10 – Organização dos modelos na API.	45
Fig. 4.11 – Modelo "PatientModel": a) função “findOne” – query simples; b) função “create” -transação.....	46
Fig. 4.12 – Organização dos middlewares e validadores na API.	47
Fig. 4.13 – Organização do NexusBRaNT.	49
Fig. 4.14 – Dialog map geral do NexusBRaNT.	50
Fig. 4.15 – Dialog map da gestão do NexusBRaNT.	51
Fig. 4.16 – Funções no ficheiro "api.js": a) registerPatient – POST; b) getPatientbyID – GET.	52
Fig. 4.17 – Componentes customizáveis no NexusBRaNT.....	53
Fig. 4.18 – Nova avaliação neuropsicológica: a) Rey15; b) IAFAI.....	54
Fig. 4.19 – Informação detalhada: a) clínica; b) programa de treino cognitivo.	55
Fig. 4.20 – Barra de navegação: a) paciente; b) PS.....	56
Fig. 4.21 – Menu: a) utilizadores em comum; b) administrador.	56

Fig. 4.22 – Formulário: a) criação; b) edição.	57
Fig. 4.23 – Funcionalidade de autenticação.	58
Fig. 4.24 – Funcionalidade de registo do paciente.	58
Fig. 4.25 – Funcionalidade de listagem de pacientes associados ao PS autenticado.	59
Fig. 4.26 – Funcionalidade de informações no paciente selecionado.	59
Fig. 4.27 – Funcionalidade da informação clínica do paciente selecionado.	60
Fig. 4.28 – Funcionalidade da informação clínica detalhada.	60
Fig. 4.29 – Funcionalidade de eliminar a informação clínica selecionada.	61
Fig. 4.30 – Funcionalidade das avaliações neuropsicológicas do paciente selecionado.	61
Fig. 4.31 – Funcionalidade de registo de uma nova avaliação neuropsicológica.	62
Fig. 4.32 – Funcionalidade de visualização dos dados normativos do instrumento "Toulouse-Piéron".	62
Fig. 4.33 – Funcionalidade da avaliação neuropsicológica detalhada selecionada.	63
Fig. 4.34 – Funcionalidade dos programas de treino do paciente selecionado.	64
Fig. 4.35 – Funcionalidade de registo de um programa de treino cognitivo.	64
Fig. 4.36 – Funcionalidade do programa de treino detalhado selecionado.	65
Fig. 4.37 – Funcionalidade do perfil cognitivo.	65
Fig. 4.38 – Funcionalidade de listas de programa de treinos: a) em curso; b) finalizados.	66
Fig. 4.39 – Funcionalidade de detalhe do programa de treino em curso selecionado.	67
Fig. 4.40 – Funcionalidade da visualização detalhada da sessão de treino detalhada.	67
Fig. 4.41 – Funcionalidade de visualização dos dados do utilizador autenticado.	68
Fig. 4.42 – Funcionalidade de edição dos dados do PS autenticado.	68
Fig. 4.43 – Funcionalidade de registo de um utilizador.	69
Fig. 4.44 – Funcionalidade de listagem dos profissionais de saúde.	69
Fig. 4.45 – Funcionalidade de visualização de dados do profissional de saúde selecionado.	70
Fig. 4.46 – Funcionalidade de listagem dos pacientes associados do PS selecionado.	70
Fig. 4.47 – Funcionalidade de associar/desassociar paciente.	71
Fig. 4.48 – Funcionalidade da visualização dos dados do administrador autenticado.	71

Índice de figuras nos anexos

Fig. A.1 – Relação de um para muitos nas tabelas "Sessions_Assessment" e "AssessmentByInstrument".	97
Fig. A.2 – Tabela "Marital Status" adicionada à BD e ligação de 1 para 1 com a tabela "SociodemographicData".	97
Fig. A.3 – Tabela "ProfessionalSituation" adicionada à base de dados e ligação de 1 para 1 com a tabela "SociodemographicData".	98
Fig. A.4 – Tabela "NormativeDataCognitive" e tabelas associadas "PatternDeviationScalarScore", "PatternDeviationInterpretation" e "ScalarScoreInterpretation".	98
Fig. A.5 – Tabela "CutoffScore" e tabela associada "NoCognitiveInterpretation".	99
Fig. A.6 – Tabela "AlarmNotification" e sua relação com a tabela "TrainingPrograms".	99
Fig. A.7 – Tabela "Training_Programs": a) antiga; b) atual.	100
Fig. A.8 – Tabela "Sessions_Training": a) antiga; b) atual.	100

Fig. A.9 – Tabela “ToulousePieronCancellationTest”: a) antiga; b) atual.....	100
Fig. A.10 – Tabela "PhonemicSemanticVerbalFluence": a) antiga; b) atual.	101
Fig. A.11 – Base de dados antiga do projeto de investigação BRaNT.....	106
Fig. A.12 – Base de dados final do projeto de investigação BRaNT.	107
Fig. B.1 – Funcionalidade de edição das informações sociodemográficas do paciente.	128
Fig. B.2 – Funcionalidade de registo de uma informação clínica.	128
Fig. B.3 – Funcionalidade de edição de uma informação clínica.....	129
Fig. B.4 - Interpretação clínica: a) cognitiva; b) não cognitiva.....	129
Fig. B.5 – Funcionalidade de edição da avaliação neuropsicológica selecionada.	130
Fig. B.6 – Funcionalidade de eliminação da avaliação neuropsicológica selecionada.	130
Fig. B.7 – Funcionalidade de eliminação do programa de treino cognitivo selecionado.	131
Fig. B.8 – Funcionalidade de edição de um programa de treino cognitivos selecionado.	131
Fig. B.9 – Funcionalidade de edição dos dados do PS selecionado.	132
Fig. B.10 – Associar/desassociar o paciente José Bruno: a) seleção do paciente; b) após ter desassociado o paciente, o mesmo aparece na seleção de associar e mensagem de sucesso da desassociação.....	133
Fig. B.11 – Funcionalidade de edição dos dados do administrador autenticado.....	133
Fig. B.12 – Funcionalidade da página não encontrada.....	134
Fig. C.1 – PowerPoint de testes de usabilidade: a) Capa; b) BRaNT; c) NexusBRaNT (definição e público-alvo); d) NexusBRaNT (principais funcionalidades); e) Agradecimento.....	135
Fig. C.2 – Questionário SUS: a) informações acerca do estudo realizado; b) consentimento e autorização; c) dados demográficos do paciente; d) questões do SUS.	138

Índice de Tabelas

Tabela 2.1 – Tabela resumo das funcionalidades dos sistemas existentes abordados....	14
Tabela 5.1 – Procedimentos dos testes de usabilidade.	73
Tabela 5.2 - Dados demográficos dos participantes.....	75

Índice de tabelas nos anexos

Tabela A.1 - Instrumento de avaliação BDI-II.....	101
Tabela A.2 - Instrumento de avaliação CDR.....	101
Tabela A.3 - Instrumento de avaliação FCSRT.....	101
Tabela A.4 - Instrumento de avaliação Figura Complexa de Rey.....	102
Tabela A.5 - Instrumento de avaliação Fluências Verbais Semânticas e Fonéticas.....	102
Tabela A.6 - Instrumento de avaliação GATSB.....	102
Tabela A.7 - Instrumento de avaliação GDS-15.....	102
Tabela A.8 - Instrumento de avaliação GDS-30.....	102
Tabela A.9 - Instrumento de avaliação PRECiS.....	102
Tabela A.10 - Instrumento de avaliação IAFAI.....	103
Tabela A.11 - Instrumento de avaliação MOCA.....	103
Tabela A.12 - Instrumento de avaliação QOLIBRI.....	103
Tabela A.13 - Instrumento de avaliação QSM.....	104
Tabela A.14 - Instrumento de avaliação Rey15.....	104
Tabela A.15 - Instrumento de avaliação SIS.....	104
Tabela A.16 - Instrumento de avaliação Toulouse-Piéron.....	104
Tabela A.17 - Instrumento de avaliação WAIS-II.....	105
Tabela A.18 - Instrumento de avaliação WHOQOL-BREF.....	105
Tabela A.19 - Instrumento de avaliação WHOQOL-OLD.....	105
Tabela A.20 - Instrumento de avaliação WMS.....	105
Tabela A.21 - Tabela "Patients" na base de dados.....	108
Tabela A.22 - Tabela "NPA" na base de dados.....	108
Tabela A.23 - Tabela "HealthProfessional" na base de dados.....	108
Tabela A.24 - Tabela "SociodemographicData" na base de dados.....	109
Tabela A.25 - Tabela "AssessmentByInstruments" na base de dados.....	109
Tabela A.26 - Tabela "MOCA" na base de dados.....	110
Tabela A.27 - Tabela "IAFAI" na base de dados.....	111
Tabela A.28 - Tabela "CDR" na base de dados.....	111
Tabela A.29 - Tabela "GATSB" na base de dados.....	112
Tabela A.30 - Tabela "GDS_30" na base de dados.....	112
Tabela A.31 - Tabela "WAIS_III" na base de dados.....	112
Tabela A.32 - Tabela "QSM" na base de dados.....	113
Tabela A.33 - Tabela "EducationLevels" na base de dados.....	113
Tabela A.34 - Tabela "WHOQOL_OLD" na base de dados.....	114
Tabela A.35 - Tabela "Rey15" na base de dados.....	114
Tabela A.36 - Tabela "WMS" na base de dados.....	115
Tabela A.37 - Tabela "ToulousePieronCancellationTest" na base de dados.....	115
Tabela A.38 - Tabela "FCSRT" na base de dados.....	116

Tabela A.39 - Tabela "FCREY" na base de dados.	116
Tabela A.40 - Tabela "PhonemicSemanticVerbalFluence" na base de dados.	117
Tabela A.41 - Tabela "WHOQOL_Bref_PT" na base de dados.	117
Tabela A.42 - Tabela " <i>SubSectionNames</i> " na base de dados.	117
Tabela A. 43 - Tabela "SIS" na base de dados.	118
Tabela A.44 - Tabela " <i>PRECiS</i> " na base de dados.	118
Tabela A.45 - Tabela " <i>Relationship</i> " na base de dados.	119
Tabela A.46 - Tabela " <i>Clinical_Infos</i> " na base de dados.	119
Tabela A.47 - Tabela " <i>Clinical_Infos_Types</i> " na base de dados.	119
Tabela A.48 - Tabela "Roles" na base de dados.	119
Tabela A.49 - Tabela "TrainingPrograms" na base de dados.	120
Tabela A.50 - Tabela " <i>Sessions_Training</i> " na base de dados.	120
Tabela A.51 - Tabela " <i>Activities</i> " na base de dados.	120
Tabela A.52 - Tabela " <i>ActivitiesVariables</i> " na base de dados.	121
Tabela A.53 - Tabela " <i>ActivitiesIterations</i> " na base de dados.	121
Tabela A.54 - Tabela "ActivitiesEvents" na base de dados.	121
Tabela A.55 - Tabela "FamilyMember" na base de dados.	121
Tabela A.56 - Tabela " <i>NPAElementDescriptor</i> " na base de dados.	122
Tabela A.57 - Tabela "CognitiveProfile" na base de dados.	122
Tabela A.58 - Tabela "GDS_15" na base de dados.	123
Tabela A.59 - Tabela " <i>MaritalStatus</i> " na base de dados.	123
Tabela A.60 - Tabela " <i>Sessions_Assessment</i> " na base de dados.	123
Tabela A.61 - Tabela "ActivitiesSettings" na base de dados.	123
Tabela A.62 - Tabela "ActivitiesSettingsIterations" na base de dados.	124
Tabela A.63 - Tabela "QOLIBRI" na base de dados.	124
Tabela A.64 - Tabela "BDI_II" na base de dados.	124
Tabela A.65 - Tabela "ProfessionalSituation" na base de dados.	125
Tabela A.66 - Tabela "NormativeDataCognitive" na base de dados.	125
Tabela A.67 - Tabela "ScalarScoreInterpertation" na base de dados.	125
Tabela A.68 - Tabela "PatternDeviationInterpertation" na base de dados.	126
Tabela A.69 - Tabela "CutoffScore" na base de dados.	126
Tabela A.70 - Tabela "NoCognitiveInterpertation" na base de dados.	126
Tabela A.71 - Tabela "PatternDeviationScalarScore" na base de dados.	127
Tabela A.72 - Tabela "InterpertationClinical" na base de dados.	127

Índice de Gráficos

Gráfico 5.1 - Resultados da questão "Acho que gostaria de utilizar este produto com frequência".....	79
Gráfico 5.2 - Resultados da questão "Considerarei o produto mais complexo do que necessário".....	80
Gráfico 5.3 - Resultados da questão "Achei o produto fácil de utilizar".....	80
Gráfico 5.4 - Resultados da questão "Acho que necessitaria de ajuda de um técnico para conseguir utilizar este produto".....	81
Gráfico 5.5 - Resultados da questão "Considerarei que as várias funcionalidades deste produto estavam bem integradas".....	81
Gráfico 5.6 - Resultados da questão "Achei que este produto tinha muitas inconsistências".....	82
Gráfico 5.7 - Resultados da questão "Suponho que a maioria das pessoas aprenderia a utilizar rapidamente este produto".....	82
Gráfico 5.8 - Resultados da questão "Considerarei o produto muito complicado de utilizar".....	83
Gráfico 5.9 - Resultados da questão "Senti-me muito confiante a utilizar este produto".....	83
Gráfico 5.10 - Resultados da questão "Tive que aprender muito antes de conseguir lidar com este produto".....	83
Gráfico 5.11 - Pontuação dos resultados do System Usability Scale.....	84

Lista de Acrónimos

AIME – *European Society for Artificial Intelligence in Medicine*

API – *Application Programming Interface*

AVC – *Acidente Vascular Cerebral*

BRaNT – *Belief Revision applied to Neurorehabilitation Therapy*

CDS – *Clinical Decision Support*

COSTAR – *COmputer STored Ambulatory Record*

CS – *Cognitive Status*

DCL – *Défice cognitivo ligeiro*

DOM – *Document Object Model*

DRY – *Don't Repeat Yourself*

EHR – *Electronic Health Record*

EMR – *Electronic Medical Records*

FCT – *Fundação para a Ciência e a Tecnologia*

HELP – *Health Evaluation through Logical Processing*

HIS – *Healthcare Information Systems*

IMIA – *International Medical Informatics Association*

IU – *Interfaces de Utilizador*

JWT – *JSON Web Token*

MIS – *Management Information Systems*

MoCA – *Montreal Cognitive Assessment*

MVC – *Model-View-Controller*

NPA – *Neuropsychological Assessment*

PAAE – *Projeto Autónomo de Automação de Escolas*

PROMIS – *Problem-Oriented Medical Information System*

PS – *Profissionais de Saúde*

RH – *Recursos Humanos*

RF – *Requisito funcional*

RT – *Requisito de tecnologia*

RNF – *Requisito não funcional*

SI – Sistema de Informação

SSL – *Secure Sockets Layer*

SUS - *System Usability Scale*

UC – *Use Case*

1. Introdução

Ao longo dos anos, os Sistemas de Informação (SI) passaram por uma grande evolução, estando os mesmos implementados em diversas áreas (como por exemplo, recursos humanos, desporto, educação, etc). Na área da saúde estes sistemas são chamados de *Healthcare Information Systems* (HIS) e os mesmos são utilizados por diversos profissionais da área.

Em 1959 foi publicado um artigo por Robert Ledley e Lee B. Lusted, onde estes abordavam que a utilização de computadores poderia automatizar a maioria do trabalho dos médicos e também evitar muitos erros humanos [1], [2]. Em 1965, nos Estados Unidos da América, foi desenvolvido um dos primeiros SI, em colaboração com a Lockheed e o Hospital El Camino [2]. Dois anos depois, em 1967 foi desenvolvido o primeiro sistema de informação hospitalar, chamado *Health Evaluation through Logical Processing* (HELP), que integrava os dados clínicos e o *clinical decision support* (CDS) [2]–[4]. Também foi criada uma organização com o objetivo de melhorar o atendimento de saúde através de ferramentas informáticas e para incentivar a evolução das ciências biomédicas e informáticas na saúde, sendo esta chamada de *International Medical Informatics Association* (IMIA) [2]. O *COmputer STored Ambulatory Record* (COSTAR) é outro sistema proposto por Octo Barnett, em 1968, sendo este um registo médico eletrónico [5]. Em 1976, foi criado o *Problem-Oriented Medical Information System* (PROMIS) a fim de armazenar os registos de cuidados de saúde na Universidade de Vermont [2]. Com a evolução destes sistemas foram criados planos de estudos focados na saúde e na informática, por exemplo, na Universidade de Stanford, foi fundado em 1980 um dos primeiros programas formais da licenciatura em informática biomédica, por Edward H. Shortliffe [6]. Após 6 anos, em 1986, foi criado a *European Society for Artificial Intelligence in Medicine* (AIME)[7], [6].

Com o crescimento destas tecnologias durante os anos 70 e os anos 80, os marcos importantes foram a transformação dos sistemas de registos em papel para sistemas informáticos, fundação de diversas associações e organizações, implementação dos princípios dos SI, segurança e proteção de dados médicos e desenvolvimento de sistemas avançados de apoio à decisão [2], [6]. Nos anos 90 houve diversos HIS implementados em alguns hospitais, ocorrendo a primeira investigação na área da telemedicina e *e-health* [2]. Também houve um enorme desenvolvimento na área de base de dados, na complexidade dos *Electronic Health Record* (EHR), na confidencialidade e na proteção de dados e normas [2], [6]. Entre 2000 e 2010, a evolução dos HIS continuou e foi possível perceber que *e-health* estava em ascensão, mas, como consequência, existiram lacunas e dificuldades na implementação (integração, interoperabilidade e avaliação da qualidade). Durante estes anos houve também a criação de subdisciplinas que envolvem a área da saúde e tecnologias, tais como: bioinformática, neuroinformática, entre outros [2], [6].

Os HIS têm como objetivo proporcionar um atendimento eficiente e de alta qualidade aos pacientes [8] [2]. Dentro dos HIS existem diversos tipos, mas os dois principais são os *electronic medical records* (EMR) e o CDS. O EMR consiste em manter as informações pessoais e médicas do paciente numa base de dados computadorizada, onde as mesmas poderão ser consultadas ao longo do tempo pelos Profissionais de Saúde (PS) autorizados [6], [9], [10]. O CDS consiste em fornecer sugestões ou notificações

adequadas para os médicos utilizarem com os seus pacientes, consoante as condições médicas dos mesmos [2], [6], [10]. As principais aplicações e funções do HIS envolvem diversos âmbitos, tais como: gestão de pacientes (registar paciente, agendar uma consulta, atualizar informação médica), gestão clínica (criar um relatório médico, enviar uma receita médica eletrónica, dar alta a um paciente), diagnóstico e tratamento (consultar exame laboratorial, criar requerimento para exame), gestão de recursos médicos (encomendar recursos, medicamentos, gerir *stock* hospitalar), gestão financeira (gerir pagamentos, controlar despesas), serviços de apoio (controlo de vacinas, controlo de infeções hospitalares) e investigação e educação (consultar biblioteca, programar horários dos funcionários, recrutamento) [6], [11].

Tal como todos os SI, os HIS, apresentam diversas oportunidades como por exemplo: reduzir custos, melhorar a qualidade de atendimento do paciente, processar e transmitir dados remotamente e reduzir erros médicos. Apesar das oportunidades, existem também desafios nomeadamente nas diferentes necessidades dos profissionais de saúde, na implementação a longo prazo, devido ao facto de existirem constantes mudanças na área da saúde, resistência na utilização destes sistemas e o elevado investimento [6], [8], [10], [12], [13].

A visualização de dados é importante nos SI devido à forma como estes são apresentados ao utilizador final. No HIS esta visualização necessita de ser bem representada para facilitar os PS que irão utilizar o sistema diretamente e, assim, analisar os dados de cada paciente.

1.1. Contexto

O trabalho desenvolvido nesta tese está enquadrado no projeto de investigação *Belief Revision applied to Neurorehabilitation Therapy* (BRaNT - Projeto PTDC/CCI-COM/30990/2017 financiado pela FCT). O projeto BRaNT consiste em criar ferramentas tecnológicas para a reabilitação cognitiva domiciliária, com auxílio de inteligência artificial, e fornecer soluções para sistemas de saúde mais sustentáveis. Tem como objetivo melhorar a qualidade de vida dos pacientes através de uma intervenção mais frequente e personalizada. Este projeto conta com a colaboração entre três universidades, a Universidade da Madeira, a Universidade de Coimbra e a Universidade Nova de Lisboa.

Uma das ferramentas a ser desenvolvida neste projeto de investigação é a NeuroAIreh@b que tem como principal objetivo a criação de um perfil cognitivo que otimiza a adaptação da prescrição através de inteligência artificial, fornecendo assim ao terapeuta várias sugestões para uma reabilitação cognitiva altamente personalizada e adaptativa através de atividades em realidade virtual baseadas na simulação de vida diária.

Na Fig. 1.1 está apresentada a arquitetura do projeto BRaNT.

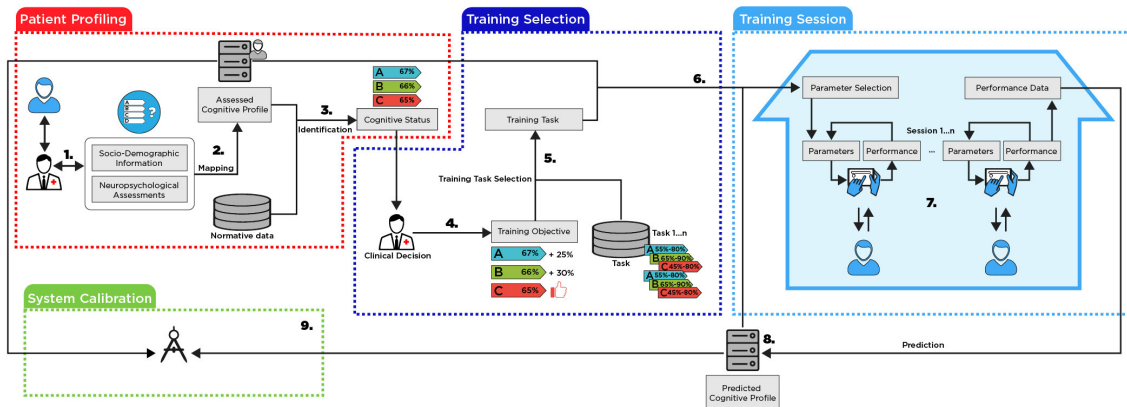


Fig. 1.1 – Arquitetura do projeto BRaNT: *Patient Profiling* – criação do perfil do paciente que integra várias avaliações neuropsicológicas para determina o estado cognitivo base; *Training Selection* – são definidos as tarefas de treino cognitivo mais apropriadas para a terapia neuropsicológica; *Training Session* – descrição das sessões de treino realizadas pelo paciente em casa; *System Calibration* – descrição da calibração que o sistema realiza comparando o perfil previsto com os dados reais do ficheiro PS recém-adquiridos.

É possível observar na Fig. 1.1 as quatro secções distinguidas por cores na arquitetura do projeto. A vermelho tem-se a secção *Patient Profiling* que é responsável pela interação entre o PS e o paciente, no qual são realizadas avaliações neuropsicológicas e obtenção de informações sociodemográficas das quais é determinado o estado cognitivo base do paciente. Em azul, *Training Selection*, o PS define os objetivos de treino do qual o paciente irá realizar remotamente em casa. O mesmo também consegue obter os parâmetros realizados pelo paciente nas sessões, prosseguindo-se assim para a secção a azul-claro (*Training Session*) no qual o paciente realiza reabilitação cognitiva domiciliária, cumprindo com o número de sessões propostas pelo PS. Cada sessão de treino é adaptada consoante a performance da sessão anterior. Os dados da performance são enviados, prevendo assim um perfil cognitivo do paciente. É realizado também um sistema de calibração (*System Calibration*) apresentado a verde.

O trabalho que será desenvolvido, no âmbito desta tese, tem como objetivo a criação de uma *dashboard* que auxilia o BRaNT nos pontos (1) e (4) da arquitetura apresentada na Fig. 1.1. O ponto (1) da arquitetura integra o componente *Patient Profiling* e tem como objetivo principal o registo da informação recolhida pelas ferramentas de avaliação neuropsicológica, em inglês *Neuropsychological Assessment* (NPA), para avaliar o funcionamento cognitivo dos pacientes. A ferramenta desenvolvida auxiliará no registo e gestão da informação sociodemográfica dos pacientes, como também na gestão das avaliações neuropsicológicas. A componente *Training Selection* do ponto (4) da arquitetura consiste em determinar objetivos de treinos específicos para cada paciente para que possa recuperar, ou compensar as capacidades cognitivas perdidas e a independência funcional baseada no *cognitive status* (CS) do paciente. Neste ponto, a ferramenta que foi desenvolvida auxilia na visualização do progresso, da performance, dos dados das sessões, na criação de um programa de treino e na apresentação das sugestões dos objetivos de treino para a reabilitação de treino de cada paciente. A partir

deste ponto e até ao final do documento, esta ferramenta será referida como NexusBRaNT.

1.2. Problema

A criação de novos HIS continua em crescimento e cada vez mais a área da saúde engloba estes sistemas. No entanto, com esta evolução, é necessário suportar os desafios apresentados anteriormente.

Os PS necessitam de uma plataforma para aceder ao *backoffice* do BRaNT e obterem todas as informações clínicas dos seus pacientes. Nesta ferramenta será necessário superar as diversas necessidades dos PS e tornar a ferramenta ajustável e completa, para que abranja as principais necessidades dos PS. Um dos problemas no desenvolvimento destas plataformas é a sua implementação a longo prazo, pois a área da saúde está em constante evolução, logo, é necessário adaptar as tecnologias desenvolvidas aos novos requisitos, como por exemplo, as necessidades de uma determinada funcionalidade e a forma como a mesma é apresentada, a descoberta de um novo diagnóstico, tratamento ou reabilitação. Um dos desafios a ser ultrapassado é a resistência na utilização de tecnologias e destas plataformas, pois nem todos os PS estão habituados a lidar com este tipo de tecnologias, apesar de estarem cada vez mais recetivos devido à evolução tecnológica atual. A segurança e a privacidade dos dados da plataforma são outro dos aspetos importantes que deverão ser garantidos, pois levantam diversas questões aos PS que irão utilizar estas plataformas.

A criação da plataforma que dá suporte aos PS, para aceder ao *backoffice*, irá facilitar o acesso, a gestão, o registo e a visualização de todas as informações dos pacientes envolvidos no projeto. Posto isto, será possível melhorar a qualidade de atendimento ao paciente e, por consequência, melhorar a qualidade de vida dos pacientes. Uma proposta de solução será apresentada no próximo ponto.

Esta plataforma foi, anteriormente, parcialmente desenvolvida no âmbito do projeto BRaNT, contudo, apresenta diversos problemas que serão solucionados neste projeto [14]. As únicas funcionalidades totalmente implementadas na plataforma anterior são: adicionar um novo paciente, adicionar novo resultado de avaliação e visualizar resultado de avaliação. Apesar destas funcionalidades estarem funcionais, o código não está organizado, desrespeitando os conceitos SOLID¹.

1.3. Solução proposta

A solução proposta divide-se em duas partes: *frontend* (plataforma *online* que os PS acedem ao *backoffice* do BRaNT) e *backend* (base de dados e desenvolvimento de uma API REST).

A primeira parte da solução para este projeto será desenvolver uma plataforma que facilite os PS acederem ao *backoffice* do BRaNT, tendo como utilizadores finais os PS. As principais funcionalidades desta plataforma são: gestão de pacientes (adicionar um

¹ SOLID,

https://web.archive.org/web/20150906155800/http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf

novo paciente, adicionar e consultar a informação clínica, visualizar a evolução do paciente, calcular a performance absoluta e a interpretação clínica), gestão de avaliações neuropsicológicas (criar uma sessão de avaliação, criar a avaliação associada a um instrumento de avaliação, visualizar os resultados de avaliações e preencher a interpretação clínica) e gestão de treinos cognitivos (criar um programa de treino cognitivo, visualizar o progresso do programa de treino cognitivo e visualizar todas as suas sessões detalhadamente). Esta solução também tem um utilizador que terá a função de administrador, no qual tem como principais funcionalidades: a criação de utilizadores, a visualização e a edição dos dados dos mesmos, a visualização dos pacientes associados e a associação e desassociação dos pacientes a um PS.

A segunda parte da solução é o desenvolvimento de uma API REST que irá responder aos pedidos tanto do *backoffice* como do NeuroAIR rehab. Estes pedidos podem ser de inserção, atualização, eliminação ou obtenção dos dados pretendidos para serem apresentados no *backoffice* ou solicitados por outros elementos do projeto, de forma a serem utilizados na plataforma referida anteriormente (NeuroAIR rehab). A API seguirá o padrão de arquitetura de *software Model-View-Controller* (MVC), do qual contém os modelos e os controladores, sendo que na mesma irá conter os *validators* para os dados. O componente *view* estará apresentado na primeira solução referida acima.

O NexusBRaNT também terá como objetivo superar os desafios apresentados no problema (ver secção 1.2) e incluir as oportunidades apresentadas anteriormente acerca dos HIS. Alguns dos desafios a serem superados são as diferentes necessidades dos PS e a resistência na utilização de sistemas tecnológicos pelos mesmos, com uma plataforma mais centrada na gestão da reabilitação cognitiva do paciente e numa navegação mais intuitiva. Uma oportunidade a incluir na plataforma é por exemplo processar e transmitir dados remotamente, através da API.

O acesso aos dados do paciente em qualquer lugar (por exemplo, casa, trabalho ou espaço público) e em qualquer equipamento (computador, *tablet*, telemóvel), permite uma maior facilidade na atualização instantânea dos dados e na monitorização do estado do paciente [12]. O desenvolvimento da plataforma também terá em conta as vantagens e desvantagens dos sistemas já existentes, que serão referidos posteriormente na revisão da literatura. Será realizado um levantamento de requisitos, juntamente com o sistema anterior, e com os PS envolvidos, seguindo as boas práticas de engenharia de requisitos e identificando assim quais as suas necessidades. A usabilidade da plataforma é outro fator importante que deverá ser abrangido, de modo a facilitar a utilização da mesma pelos PS. Os princípios da programação, padrões de desenho e regras serão essenciais para implementação correta e completa do código.

Deste modo, o sistema desenvolvido será uma aplicação *web* destinada aos PS de forma a auxiliar os mesmos na gestão da reabilitação e da avaliação neuropsicológica dos seus pacientes. O NexusBRaNT terá uma base de dados (parcialmente já desenvolvida) que irá conter toda a informação armazenada do *backoffice* do projeto BRaNT e a mesma terá de ser consultada para aquisição de dados e visualização dos mesmos no *website*.

1.4. Estrutura do relatório

O relatório está dividido em 7 capítulos, começando pela introdução, onde serão apresentados os HIS, o contexto, o problema e a solução proposta.

A revisão de literatura é apresentada no segundo capítulo. É apresentada informação acerca do défice cognitivo ligeiro, as várias alternativas já existentes ao sistema implementado, as ferramentas de desenvolvimento que existem, as seleccionadas para o desenvolvimento deste projeto e uma breve conclusão do capítulo.

No terceiro capítulo está apresentada toda a metodologia utilizada, sendo que a mesma se divide em arquitetura proposta, requisitos, casos de utilização e método de desenvolvimento.

A implementação está apresentada no quarto capítulo. Os subcapítulos desta secção são: as ferramentas auxiliares, a base de dados e todo o desenvolvimento do *backend* e *frontend*.

A avaliação do sistema foi dividida em teste do sistema e avaliação do mesmo, sendo estes apresentados no quinto capítulo. É neste onde estão todos os testes de usabilidade realizados após o desenvolvimento da plataforma.

No sexto capítulo é apresentado a discussão de resultados que provém dos testes realizados aos utilizadores e o *feedback* das *experts* do sistema.

Por último tem-se o sétimo capítulo onde está apresentado a conclusão deste projeto e as sugestões de trabalhos futuros.

2. Revisão de literatura

Neste capítulo é apresentada a revisão de literatura começando com a definição, o diagnóstico e as recomendações para prevenção do déficit cognitivo ligeiro. Em seguida serão apresentados sistemas existentes que envolvam SI na área da saúde e em outras áreas e também a revisão da plataforma anteriormente desenvolvida para o *backoffice* do BRaNT. Também serão referidas ferramentas de desenvolvimento que já existem em *backend*, *frontend* e em base de dados, como também as ferramentas selecionadas para a implementação da solução proposta.

2.1. Déficit cognitivo ligeiro (DCL)

A evolução das tecnologias e da utilização das mesmas podem integrar diversas áreas da saúde. Existem soluções desenvolvidas para a prevenção e controlo desta condição patológica como também a gestão dos pacientes. O NexusBRaNT será focado na gestão dos programas de treinos, avaliações neuropsicológicas e monitorização em tempo real do paciente.

O DCL foi definido em 1997 como uma deficiência clínica e neuropsicológica, considerada como um estado intermédio entre o envelhecimento normal e a demência. Atualmente é considerado como uma condição patológica devido a ser potencialmente progressivo [15].

Esta condição patológica não interfere significativamente nas atividades básicas do dia-a-dia dos pacientes. No entanto, quando as atividades têm maiores exigências do ponto de vista cognitivo, poderão interferir na execução destas atividades instrumentais diárias (por exemplo, executar funções profissionais, gestão de assuntos financeiros, entre outras) [16], [17].

O diagnóstico de DCL pressupõe a exclusão de vários fatores e condições médicas que poderão explicar a sintomatologia cognitiva que lhe é característica [15]–[17]. A avaliação neuropsicológica consiste num procedimento padrão que envolve a gestão de instrumentos de avaliação das funções nervosas superiores que visa contribuir para o diagnóstico e monitorização clínica de quadros clínicos neurológicos, como é o exemplo do DCL [17]–[19].

2.2. Sistemas existentes

Em seguida serão apresentados sistemas existentes que envolvam a gestão dos pacientes e também a reabilitação cognitiva através da *web*. Também serão referidos alguns SI utilizados noutras áreas para gestão. É fundamental fazer uma revisão de sistemas semelhantes ao NexusBRaNT, para analisar o que já existe, quais são as suas vantagens e principalmente para entender as limitações existentes. Ainda neste subcapítulo será apresentada uma discussão acerca do antigo sistema do *backoffice* do BRaNT, referindo os pontos fortes e fracos do mesmo.

Para obter todas as informações sobre os sistemas de gestão de pacientes e reabilitação cognitiva recorreu-se primeiramente ao teste das plataformas, sendo que a plataforma CogiFit [20] permitiu testar algumas das suas funcionalidades de forma gratuita, como

por exemplo: jogos de memória, coordenação, atenção, entre outros. As principais funcionalidades semelhantes ao NexusBRaNT nas outras plataformas eram todas com plano de subscrição pago. Deste modo, teve-se de recorrer à documentação disponibilizada nos *websites* de cada plataforma ([20], [21], [22], [23]) de forma a poder obter toda a informação acerca das mesmas.

2.2.1. Sistemas de reabilitação cognitiva

Alguns dos sistemas utilizados para gestão de pacientes e reabilitação cognitiva são: COGWEB [20], CogniFit [21], NeuronUp [22] e RehaCom [23], sendo estes abordados mais aprofundadamente em seguida. Também existem sistemas como HappyNeuron Pro [24] que têm objetivos e funcionalidades semelhantes aos apresentados em seguida. Estes sistemas foram selecionados tendo em conta a investigação realizada acerca da reabilitação cognitiva, e também com a experiência profissional das PS do projeto de investigação.

2.2.1.1. COGWEB

O COGWEB é um sistema que permite os pacientes realizarem treinos cognitivos com acompanhamento de um PS. Pode ser utilizado individualmente ou em grupo, presencialmente ou remotamente e em instituições ou em comunidade. Os treinos cognitivos têm diferentes exercícios em domínios cognitivos, como por exemplo: atenção, memória, cálculo, funcionamento executivo e linguagem. Os exercícios contêm diferentes dificuldades e as mesmas são personalizadas automaticamente consoante o progresso do paciente. Existem pré-requisitos obrigatórios para o sistema por parte do paciente, como o diagnóstico médico e a caracterização neuropsicológica pormenorizada das deficiências cognitivas e, por parte do PS, como a disposição para gestão do programa de tratamento. Os pacientes e PS têm acesso à ferramenta, sendo que ambos tem funcionalidades diferentes [25], [26].

Os PS acedem à plataforma através das suas credenciais (*username* e *password*) e têm acesso à informação dos seus pacientes. O menu principal do PS é o registo de saúde do paciente, que contém cinco submenus com formulário médico (dados de identificação, dados clínicos e duração da licença de utilização do sistema e as credenciais dos pacientes para aceder), planos de intervenção (registo da informação geral, da duração, dos principais domínios cognitivos e a intensidade esperada), programação das sessões (criação das sessões em modo padrão ou modo avançado, escolha dos exercícios, datas de início e conclusão, ordenação dos exercícios, duração de cada exercício e definição do nível inicial e final esperado) e resultados (apresentação de vários tipos de gráficos com o progresso, respostas certas vs erradas, níveis concluídos, tempo de cada jogo, entre outros) [25], [26].

Na Fig. 2.1 estão apresentados exemplos dos gráficos dos resultados e do progresso na plataforma COGWEB.

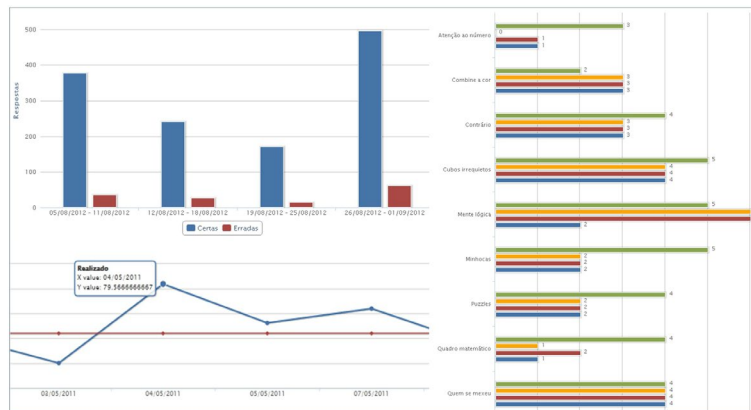


Fig. 2.1 – Exemplos dos gráficos de progresso na plataforma COGWEB [26].

A plataforma no lado dos pacientes está dividida em três fases: sessões de treino, *game arena* e ferramentas de motivação. Quando o paciente inicia sessão na plataforma, é apresentado um ecrã com as informações da sessão de treino e um botão para iniciar. Em seguida, é apresentada a *game arena*. Para cada exercício é exibido um conjunto de instruções (por escrito e por áudio) com duração de 20 segundos e posteriormente o exercício começa. Durante os jogos, o paciente pode visualizar o tempo restante, o botão pausa, o botão de ecrã inteiro, o botão para avançar para o próximo jogo, nome do exercício, o nível e a resposta (certa ou errada). No fim de cada nível bem-sucedido é exibida uma mensagem de suporte no ecrã e por áudio, caso contrário não é apresentada nenhuma mensagem ao paciente e as instruções do próximo exercício são exibidas [25] [26]. No NexusBRaNT não terá o lado correspondente aos pacientes porque o mesmo está a ser desenvolvido numa aplicação no Unity3D separadamente.

Na Fig. 2.2 são apresentados alguns exemplos de exercícios realizados pelos pacientes, sendo estes “encontre o número” e “rápida memória”.



Fig. 2.2 – Alguns exemplos de exercícios realizados pelos pacientes na plataforma COGWEB [21].

2.2.1.2. CogniFit

O CogniFit é uma plataforma que suporta testes cognitivos (avaliação cognitiva geral, condução, concentração, percepção, memória, raciocínio, entre outros), treinos cognitivos (para adultos e crianças com diversas habilidades cognitivas, sendo estas: a memória contextual, a atenção dividida, a denominação, a inibição, a percepção visual, entre outras),

ferramentas para profissionais (médicos e psicólogos, escolas e investigadores). O lado da ferramenta direcionada para os PS serve para ajudar na deteção, no diagnóstico e na intervenção dos transtornos cognitivos [21].

Os PS podem realizar testes cognitivos com padrões para diagnosticar, e monitorizar os pacientes consoante a reabilitação cognitiva. É possível também os PS adicionarem e gerirem a informação dos pacientes (Fig. 2.3 b)), como também programarem os treinos cognitivos (Fig. 2.3 a)). Os treinos cognitivos são adaptados automaticamente consoante as necessidades de cada paciente [21].

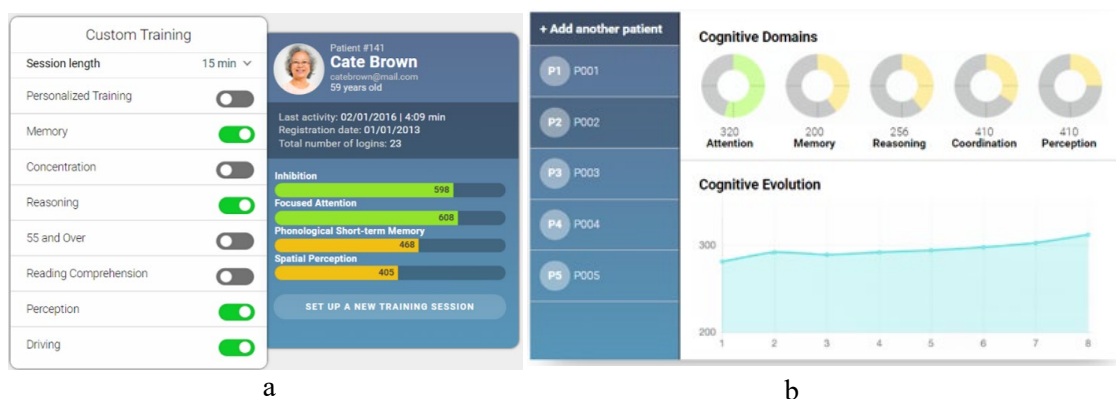


Fig. 2.3 – Interface do CogniFit: a) programação da sessão de treino; b) gestão do paciente e visualização gráfica dos resultados [22].

2.2.1.3. NeuronUp

O NeuronUp é uma plataforma *web* de reabilitação e estimulação cognitiva que dá suporte aos PS. A reabilitação e estimulação cognitiva é abordada em algumas áreas tais como, lesão cerebral adquirida, doenças neurodegenerativas, deficiência intelectual, doença mental (esquizofrenia e transtorno bipolar) e envelhecimento normal. A plataforma contém diversos materiais e recursos que permitem a criação de sessões de treino e também a gestão da informação dos pacientes e dos resultados das sessões [22].

Esta plataforma consiste em duas seções, a gestão de atividades e a gestão de resultados, tendo as seguintes características [22]:

- **Gestão de atividades** (Fig. 2.4 a)): Contém diversos exercícios, ferramentas e simuladores que permitem que o paciente realize funções cognitivas básicas, atividades do dia-a-dia e habilidades sociais. As atividades são classificadas consoantes as áreas de intervenção, os graus de deficiência, realização individual ou em grupo, nível sociocultural ou educacional, idade e limitações sensoriais. Os PS têm possibilidade de escolher e personalizar as atividades das sessões de treinos com o objetivo de adaptar as mesmas consoante as necessidades de cada paciente. É possível realizar as atividades manualmente (lápiz e papel) ou digitalmente (computador, telemóvel ou *tablet*);
- **Gestor de resultados** (Fig. 2.4 b)): Os PS têm acesso à organização, ao controlo e à consulta das atividades realizadas pelos pacientes. Quando as atividades são realizadas digitalmente a plataforma guarda automaticamente os resultados quantitativos. É possível guardar na plataforma os resultados das

atividades realizadas manualmente. A plataforma permite aos PS adicionarem observações qualitativas para complementar a informação.

Na Fig. 2.4 a) b) estão representadas as interfaces da plataforma NeuronUP.

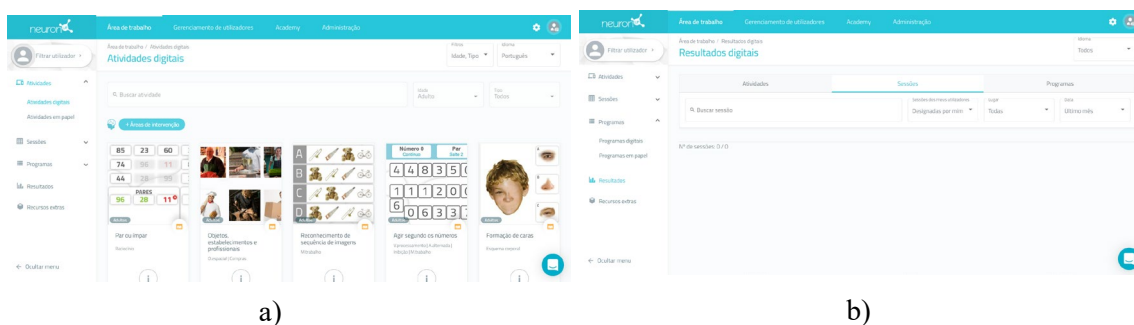


Fig. 2.4 – Interface do NeuronUp: a) gestão de atividades; b) gestão de resultados [23].

2.2.1.4. RehaCom

O RehaCom é outra plataforma que permite medir, treinar e monitorizar reabilitação cognitiva através do computador. A plataforma pode ser utilizada por neuropsicólogos, terapeutas ocupacionais e clínicos que trabalham em hospitais, clínicas ou centros de reabilitação. Atualmente é possível os pacientes realizarem o treino em casa, sendo controlado e monitorizado pelo profissional. Existem então 2 tipos distintos de utilizadores na plataforma [23]:

- **Clínicos:** existe uma funcionalidade de triagem com diversos módulos para deteção de deficiências. Consoante o diagnóstico do paciente a plataforma recomenda alguns módulos de treino cognitivo necessários para a sua reabilitação e o profissional pode também definir os parâmetros e módulos para o treino. A plataforma também permite uma triagem regular para apresentação da progressão do paciente e relatórios detalhados sobre o mesmo. Quando a intervenção é remota o profissional de saúde pode prescrever um programa de treino cognitivo e acompanhar remotamente os seus pacientes. Toda evolução dos pacientes e resultados dos treinos cognitivos podem ser consultados pelos PS (Fig. 2.5 a)).
- **Pacientes:** A plataforma adapta-se às capacidades cognitivas e organiza as sessões de treino cognitivo consoante a performance do utilizador, tendo como objetivo a progressão contante e redução da frustração do paciente ao realizar as sessões. O programa de treino consiste na realização de tarefas cognitivas (Fig. 2.5 b)). A plataforma pode ser adaptativa consoante o paciente, o nível dos *déficits* visuais ou estímulos familiares (usar imagens da família nos treinos). O sistema também permite que os pacientes continuem a realizar os módulos mesmo fora das suas sessões de treino.

Na Fig. 2.5 a) e b) estão apresentados os ecrãs dos resultados do módulo atenção e concentração que o PS têm acesso e de um treino do mesmo módulo, respetivamente.

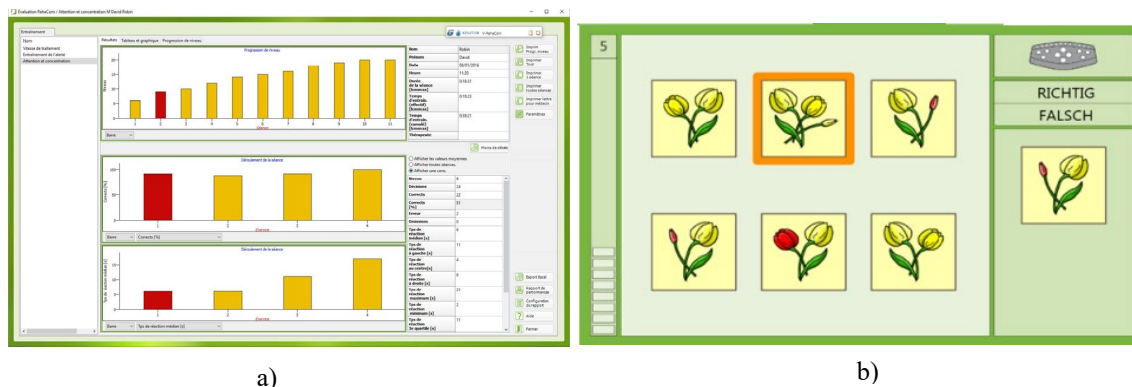


Fig. 2.5 – Interface do RehaCom - módulo atenção e concentração: a) resultados; b) treino cognitivo [24].

2.2.2. Noutras áreas

Para além da área da saúde é possível encontrar outros sistemas desenvolvidos que têm como objetivo auxiliar na gestão de pessoas e de recursos. Estes sistemas desenvolvidos são SI que têm semelhanças com os sistemas anteriormente apresentados. Um SI dá suporte à transmissão de informação internamente e externamente de uma organização. Existem 5 funções principais de um SI, sendo estas: recolha, armazenamento, processamento, representação e distribuição de informação [27]. Os *management information systems* (MIS) são SI que abordam a gestão nas organizações, tendo como funções: planeamento, controlo e tomada de decisão das informações. Existem diversos tipos de aplicações onde os MIS são introduzidos, tais como: gestão de vendas, controlo de *stock*, recursos humanos (RH), orçamento anual e entre outros [27]. Serão apresentados nos próximos parágrafos alguns exemplos de SI nas áreas de desporto, da educação e da gestão de RH.

Na área da educação existem sistemas que auxiliam na gestão da informação escolar, como por exemplo o Classter [28] e o Projeto Autónomo de Automação de Escolas (PAAE) [29]. O Classter contém aplicações *web* destinadas à educação, tendo como principais portais *web*: aluno, professor, pais e funcionário. No portal do aluno é possível consultar as informações escolares, as faltas, o calendário e horário escolares, anúncios, trabalhos, comentários e a biblioteca escolar. Também são fornecidos os horários e os trajetos dos transportes públicos, cursos da plataforma *Moodle* e acesso a aplicações educacionais externas. O portal do professor permite o acesso às informações dos seus alunos, à gestão de eventos (trabalhos de casa, questionários, avaliações, testes e observações), ao atendimento, ao registo de comportamento, à criação de bibliotecas de cursos e aos documentos. O portal dos pais consiste no acesso das informações dos seus filhos (horário e desempenho escolar), visualização dos trabalhos, notas, faltas de comparecimento e comunicação entre pais e professores. O portal do funcionário é direcionado para o quem trabalha com o *backoffice* escolar. No mesmo é possível aceder, gerir e modificar informações acerca de registos, matrículas, serviços académicos e financeiros [28]. O PAAE é um projeto que consiste em criar soluções tecnológicas que auxiliam a gestão escolar. Este está dividido em 12 módulos: pedagógico (gestão de matrículas e turmas, importação de horários, consultar/exportar dados e estatísticas

diversas), portal professor (gestão de sumários, marcação de testes e ausências, gestão de avaliações e monitorização de aulas), portal de aluno/encarregados de educação (consultar faltas, horário, entradas/saídas, trabalhos de casa e folha de vencimento, renovação da matrículas e gestão de refeições), faturação multiempresa, fornecedores/*stock*, RH e vencimentos, ação social escolar, refeitório, *buffet*, papelaria e reprografia (controlo de vendas), quiosque e controlo de acessos [29].

Tal como nas áreas anteriores, para os RH, também existem plataformas que simplificam toda a gestão que envolve os mesmos. A Factorial [30] é um *software* que suporta a gestão de RH através de algumas funcionalidades, sendo estas: férias e ausências (ausências personalizadas, supervisor de ausências, calendário de equipa e política de férias), sistema de controlo de assiduidade e gestão de turnos (relatórios personalizados, registo de assiduidade e aplicação de relógio de ponto), relatórios e indicadores de RH (relatório personalizados, painel de relatórios e organograma) e gestão documental (recibos de vencimentos, contratos e documentos da empresa) [30]. O uMan Global HR [31] é uma plataforma completa e especializada para RH com diversas funcionalidades tais como: processamento salarial, gestão de presenças, de desempenho, de competências, de formação e da cantina, controlo de produtividade, orçamentação, portal do colaborador e controlo da segurança e saúde no trabalho [31].

2.2.3. Pontos fortes e fracos do sistema existente

A plataforma anteriormente desenvolvida para ser *backoffice* do BRaNT contém diversos pontos fortes e fracos que necessitam de ser melhoradas e ultrapassadas, respetivamente. De forma a analisar os pontos fortes e fracos da plataforma foi necessário executar o código da mesma de forma a poder interagir com o sistema. Algumas das funcionalidades desenvolvidas são comuns aos sistemas referidos anteriormente, que reflete que houve um estudo prévio do que já existia e o mesmo foi adaptado para a plataforma. Existiu também a preocupação de incluir algumas funcionalidades novas, como por exemplo: visualização das ferramentas de avaliação e a inserção de novas. Todos os requisitos da plataforma necessitam de ser revistos e corrigidos juntamente com os PS envolvidos, de forma a obter descrição detalhada e precisa do sistema, sendo que posteriormente no capítulo 3.1 serão apresentados os novos requisitos elicitados e organizados.

As funcionalidades implementadas e que estão funcionando corretamente são: adicionar um novo paciente, adicionar novo resultado de avaliação e visualizar resultado de avaliação. Apesar do funcionamento das mesmas é necessário melhorar, organizar e transformar o código de forma que respeite as boas práticas (SOLID) e a arquitetura de *software* escolhida.

É preciso desenvolver as restantes funcionalidades essenciais às PS do projeto e adaptar alguns dos termos utilizados nas anteriores, como por exemplo usar o termo de “instrumento de avaliação” em vez de “ferramenta de avaliação”, indo ao encontro dos requisitos que serão levantados com os PS.

O *design* da plataforma também necessita de ser adaptado, de modo a facilitar a usabilidade dos PS com a plataforma, sendo que alguns dos ecrãs tiveram como base o *design* anterior, mas os mesmos serão referidos na secção 4.4.3.

A avaliação do sistema antigo e a seleção destes pontos tiveram como base o *feedback* de elementos do projeto, pois estes colegas utilizaram o sistema do ponto de vista do utilizador e do de desenvolvimento. O sistema antigo também foi avaliado através de testes realizados no âmbito desta tese.

2.2.4. Análise comparativa dos sistemas estudados

Com base na revisão realizada anteriormente dos diversos sistemas existentes de gestão dos pacientes e reabilitação cognitiva é possível identificar semelhanças entre os sistemas. Na Tabela 2.1 está apresentada um resumo das funcionalidades em comum nos diversos sistemas abordados na secção 2.2.1.

Tabela 2.1 – Tabela resumo das funcionalidades dos sistemas existentes abordados.

		Sistemas existentes abordados			
		COGWEB	CogniFit	NeuronUp	RehaCom
Funcionalidades	Gestão de pacientes	✓	✓	✓	✓
	Gestão de avaliações neuropsicológicas	✗	✗	✗	✗
	Gestão de programas de treinos	✓	✓	✓	✓
	Jogos/tarefas cognitivas	✓	✓	✓	✓
	Adaptação da dificuldade nos treinos (automática)	✓	✓	✓	✓
Tipo de utilizadores	Pacientes	✓	✓	✓	✓
	Profissionais de saúde	✓	✓	✓	✓
	Investigadores	✗	✓	✓	✗
	Alunos	✗	✓	✓	✗
	Professores	✗	✓	✓	✗

Observando a Tabela 2.1 verifica-se que os sistemas apresentados contêm a reabilitação cognitiva através de treinos que incluem jogos e atividades cognitivas e também monitorização da mesma e dos pacientes. Também é possível visualizar a evolução cognitiva do paciente através dos resultados da reabilitação que é controlada e monitorizada pelos PS. A adaptação da dificuldade dos jogos/tarefas cognitivas é automática em todos os sistemas consoante a performance do paciente.

O NexusBRaNT tem algumas das características presentes nos sistemas já existentes, mas terá como público-alvo apenas os PS, sendo possível abranger todas as necessidades e melhorar a usabilidade para os PS. Posto isto, foram desenvolvidas mais profundamente as funcionalidades de programação dos treinos cognitivos, avaliações neuropsicológicas, instrumentos de avaliação, gestão da informação do paciente (sociodemográfica e clínica), entre outras.

Foram abordados neste capítulo outros SI que envolvem áreas distintas (desporto, educação e RH) que têm como objetivo auxiliar na gestão de pessoas e de recursos, sendo possível identificar semelhanças entre estes sistemas e os de gestão dos pacientes e reabilitação cognitiva. Por exemplo, na gestão de treinos (desporto), na gestão de eventos (educação), na gestão férias e ausências (RH), na gestão da informação dos atletas, dos alunos e dos funcionários e na gestão de documentação com diversas semelhanças, como a programação, a modificação, a calendarização e a gestão de informação. Estas

funcionalidades aproximam-se assim também de algumas funcionalidades dos sistemas de reabilitação cognitiva e do NexusBRaNT (funcionalidades apresentadas na secção 2.2.1 e 1.3, respetivamente).

Os pontos referidos acerca da plataforma anteriormente desenvolvida serão essenciais para o desenvolvimento da nova plataforma. Deverá ser melhorado o que já estava funcional, desenvolver as diversas funcionalidades referidas em falta e melhorar o *design*, sempre tendo atenção as necessidades dos PS e as boas práticas de *software*. Como por exemplo, na visualização da lista dos pacientes o componente da tabela personalizada deverá estar separado apenas num ficheiro e sempre que necessário o utilizar é executado o componente na página pretendida. Deste modo evita-se a redundância de código e a importação de todos os componentes utilizados no componente de tabela, sempre que a mesma é executada.

2.3. Ferramentas tecnológicas

Em seguida serão expostas as ferramentas existentes que auxiliam no desenvolvimento e implementação de sistemas, e que foram consideradas para o desenvolvimento do sistema proposto. As ferramentas de desenvolvimento revistas serão divididas em 3 categorias: *backend*, *frontend* e base de dados. Na categoria *backend* são abordadas as seguintes tecnologias: Laravel [32], CakePHP [33], Django [34], Spring [35] e Express.js [36]. As ferramentas *frontend* que serão abordadas são: Vue.js [37], Angular [38], Ember [39], jQuery [40] e React [41]. Serão também expostas as categorias de base de dados existentes, relacionais e não relacionais.

Para a obtenção das informações abaixo descritas recorreu-se à pesquisa das ferramentas mais utilizadas atualmente e à consulta da documentação presente em cada ferramenta descrita.

2.3.1. Backend

O *backend* de um *website* ou programa de *software* são todos os programas e *scripts* que executam nos servidores, tornando assim as páginas *web* dinâmicas e interativas. Existem um conjunto de tarefas que ocorrem no *backend*, como por exemplo: processamento de formulários, gestão da base de dados, sistemas de gestão de conteúdos e utilização de outras aplicações *web* utilizando PHP, JSP, Ruby, ASP.NET, entre outras linguagens de programação. O utilizador final não tem acesso a este lado da aplicação [42].

Laravel é uma *framework* que usa a linguagem de programação PHP, onde é possível desenvolver aplicações *web* através de uma arquitetura *Model-View-Controller* (MVC). Foi desenvolvida para realizar suposições inteligentes do que está a ser implementado para que o utilizador chegue ao seu objetivo em menos linhas de código, mas nem todas as aplicações desenvolvidas poderão utilizar estas suposições inteligentes. Tem como vantagem a modularidade, através de *drivers* e do *bundles system*, que auxiliam a modificação e o aumento da *cache*, da sessão, da base de dados, da funcionalidade da autenticação e da reutilização de código. É ainda possível a interação (migrações) com as bases de dados, devido ao conjunto de ferramentas fornecidas por esta *framework* [32], [43].

Uma *framework open source* também utilizada é o CakePHP que utiliza a linguagem programação PHP e a arquitetura MVC. Contendo alguns recursos como a compatibilidade com o PHP 4 e PHP 5, interação com a base de dados (utilizando a funcionalidade *create, read, update e delete* - CRUD), *templates* rápidos e flexíveis, flexibilidade com a *cache*, geração de código, limpeza de dados e utilização de componentes (por exemplo, email, *caches*, segurança e sessões) [33].

Outra *framework* muito popular é o Django que utiliza a linguagem de programação Python e, como a anterior, contém uma arquitetura MVC do lado do servidor. Inicialmente, a mesma só funcionava com bases de dados relacionais, sendo que atualmente opera com bases de dados não-relacionais através de pacotes externos. O Django segue alguns princípios de *design*, sendo estes: o princípio *Don't Repeat Yourself* (DRY), o princípio *Explicit Is Better Than Implicit* e a arquitetura *Loosely Coupled*. É escalável devido à rapidez e flexibilidade da transmissão de dados [34], [44].

Outra alternativa é *framework* Spring que utiliza como linguagem de programação o Java. Tal como as anteriores, esta fornece uma estrutura MVC. No desenvolvimento de aplicações é possível recorrer a extensões para a implementação de aplicações *web* utilizando a plataforma Java, sendo assim uma *framework* flexível. A gestão de transações declarativas, acesso remoto às mesmas através de interface de programação de aplicações ou serviços *web*, e persistência dos dados são algumas das funcionalidades suportadas pela Spring [35], [45].

O Express.js é uma *framework open source* baseada no módulo Node.js HTTP e nos componentes *Connect*, sendo estes componentes chamados de *middlewares*. Os programadores conseguem selecionar as bibliotecas que necessitam para desenvolvimento de um projeto, tendo flexibilidade e personalização elevada. Esta *framework* resolve problemas relacionados com a repetição de código em tarefas semelhantes (por exemplo, *parsing* de *cookies*, gestão de sessões e organização de *routes* com uma sequência de condições baseadas em caminhos URL e métodos HTTP dos pedidos) em aplicações onde são utilizadas apenas módulos do Node.js. O Express.js pode seguir a arquitetura MVC, sendo este um padrão de arquitetura de *software*, para o desenvolvimento de aplicações *web* [46]. O Node.js é uma plataforma de *software* utilizada para implementar aplicações de redes escaláveis usando JavaScript [36].

2.3.2. Frontend

O *frontend* envolve todo o processo de desenho e de apresentação no *browser*, sendo possível as seguintes tarefas: desenho da interface, implementação de documentos HTML e *style sheet*, implementação lógica (por exemplo, JavaScript), produção de imagem e desenho gráfico. O utilizador final tem acesso direto a estas [42].

O Vue.js é considerado um *progressive framework* para o desenvolvimento de interfaces de utilizador (IU). A biblioteca principal desta *framework* aborda essencialmente a *view layer* que torna fácil adaptação e integração com outras bibliotecas ou projetos já desenvolvidos. Suporta aplicações *single-page* sofisticadas através do auxílio e combinação de ferramentas modernas e bibliotecas de apoio [37].

Outra plataforma de desenvolvimento é o Angular que foi construída em *TypeScript*. Esta consiste numa estrutura baseada em componentes que permite a implementação de

aplicações *single-page*, aplicações móveis e aplicações *desktop* eficientes e sofisticadas com a reutilização de código. Contém *routing*, gestão de formulários e comunicação cliente-servidor através de bibliotecas integradas [38].

Uma *framework* JavaScript popular é o Ember.js que auxilia na implementação de aplicações *web* modernas. Tal como as *frameworks* anteriores, esta é baseada em componentes, possibilitando a reutilização de elementos na IU. O Ember.js também contém *routing*, *services*, *Ember Data* (consistência na comunicação de APIs externas), *Ember Inspector* (extensão no *browser* para inspecionar aplicação em tempo real) e *Ember CLI* (*toolkit* de desenvolvimento que permite a criação de aplicações Ember) [39].

Outra biblioteca JavaScript utilizada é jQuery que disponibiliza diversos recursos, sendo alguns deles: a manipulação de HTML/DOM, de CSS, de eventos, efeitos, animações e Ajax, tendo compatibilidade com diversos *browsers*. É considerada uma biblioteca versátil e extensível [40].

O React é outra biblioteca JavaScript que permite criar IU baseada em componentes reutilizáveis. Através da criação destes componentes é possível fazer composição dos mesmos para criação de IU complexas. É considerada uma biblioteca declarativa, eficiente e flexível, visto que a mesma irá atualizar e renderizar apenas os componentes necessários conforme a atualização dos dados. É possível utilizar também o *Document Object Model* (DOM) na implementação de aplicações robustas e de elevada performance [41].

2.3.3. Base de dados

A base de dados é um conjunto de dados que podem ser mantidos, atualizados, removidos ou consultados (pesquisados), sendo um componente crucial para qualquer tipo de sistema de informação [47]. Existem dois tipos principais de bases de dados: relacional e não relacional. De uma forma geral, as bases de dados não relacionais têm uma vantagem a nível de escalabilidade e flexibilidade devido à fácil inserção de novos dados e à grande quantidade de modelos de dados, enquanto as bases de dados relacionais são fortes na organização de dados complexos e altamente conectados.

Uma base de dados relacional é um conjunto de tabelas com um nome único e cada uma guarda instâncias de uma entidade e/ou associação. Este tipo de base de dados pode conter múltiplas relações, pois o armazenamento de todas as propriedades numa única relação resulta em informações repetidas e a existência de valores nulos [47]. É utilizada a *Structured Query Language* (SQL) para o acesso e modificação dos dados na base de dados. O MySQL e o Oracle são dois exemplos de bases de dados relacionais [48].

A base de dados não relacional também armazena dados, mas não contém tabelas, linhas, chaves primárias ou estrangeiras, assim sendo não existem relações entre os dados. É utilizado um modelo otimizado de armazenamento para dados específicos. Como base de dados não relacionais, existe o MongoDB e o DynamoDB como exemplos [48], [49].

2.3.4. Ferramentas selecionadas

As ferramentas de desenvolvimento escolhidas para a implementação do NexusBRaNT foram a *framework* Express.js para *backend*, a biblioteca React para *frontend* e base de dados relacional, mais concretamente o MySQL, para armazenamento

dos dados. A seleção das ferramentas foi com base nas características das mesmas e nas vantagens em relação às outras ferramentas anteriormente apresentadas.

O Express.js e o React têm como linguagem de programação o JavaScript, o que auxiliará na aprendizagem mais rápida, e implementação do NexusBRaNT, sendo possível utilizar o Node.js em ambos. A utilização de uma arquitetura de *software* MVC, a resolução dos problemas apresentados anteriormente no Express.js, a utilização de componentes reutilizáveis no React e a reutilização código em ambos permitirá o desenvolvimento de uma aplicação eficiente, flexível e robusta.

Uma base de dados relacional é fundamental neste projeto, pois existem diversas relações entre os dados que serão utilizados e assim será possível uma melhor gestão dos mesmos. Um exemplo de uma relação complexa no projeto é uma avaliação neuropsicológica que envolve o paciente, o PS e os instrumentos de avaliação que serão utilizados na mesma. É necessário relacionar na avaliação neuropsicológica quais os instrumentos que serão utilizados, o paciente que terá a intervenção e o PS que irá realizar a avaliação neuropsicológica. O facto de algumas tabelas existentes na base de dados serem fixas e dificilmente serão alteradas contribuiu para seleção deste tipo de base de dados como, por exemplo, o instrumento de avaliação *Montreal Cognitive Assessment* (MoCA) que teve a última validação realizada entre 2003 e 2004 [50].

3. Especificação do problema

Neste capítulo do trabalho é apresentada a metodologia que foi utilizada para a solução do sistema incluindo o levantamento de requisitos, a arquitetura proposta, o método de avaliação e o planeamento das tarefas. Na implementação do NexusBRaNT foi seguido o método ágil para um desenvolvimento incremental, comunicação contínua com os PS e flexibilidade nas alterações necessárias.

A abordagem realizada neste trabalho seguiu as boas práticas nas diferentes etapas de engenharia de requisitos [51]. Será realizada a elicitación, análise, especificación, priorización e validación dos requisitos. Os requisitos foram elicitados juntamente com as especialistas que integram o projeto BRaNT.

3.1. Requisitos

Os requisitos de um sistema são a descrição de como este se deve comportar, uma propriedade ou atributo do mesmo. Um sistema de *software* contém três níveis diferentes de requisitos, sendo estes: requisitos de negócio, de utilizador e funcionais [51]. Na Fig. 3.1 está apresentado o processo geral realizado nos requisitos.

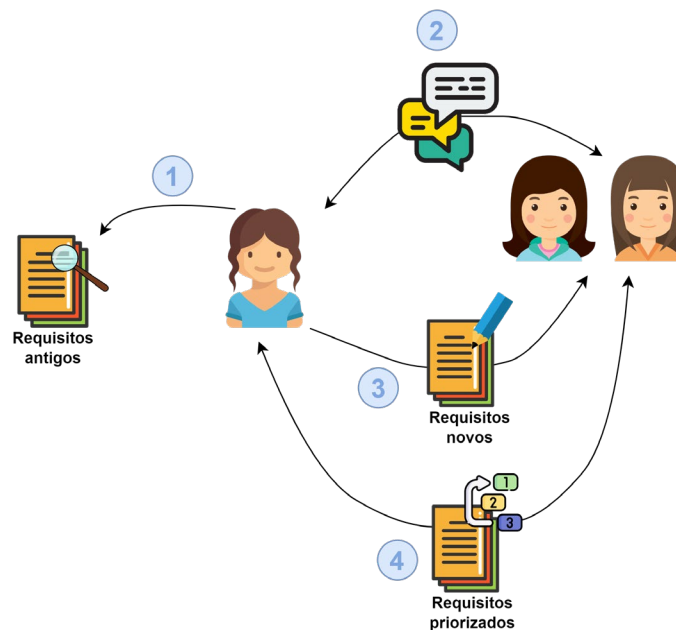


Fig. 3.1 – Processo geral dos requisitos.

Neste trabalho foram revistos os requisitos funcionais do antigo sistema apresentado na secção 2.2.3. Começou-se por analisar os requisitos (ponto 1 da Fig. 3.1), ordenando-os por funcionalidades, visto que foram detetadas algumas incoerências entre os requisitos, por exemplo: a repetição dos mesmos, a falta de organização e de categorização por funcionalidades. Após esta análise inicial foi necessária uma reunião com as PS responsáveis para uma nova elicitación de requisitos e correção dos requisitos anteriormente elicitados (ponto 2 da Fig. 3.1). As PS que colaboram no levantamento dos requisitos são duas das investigadoras do projeto BRaNT, sendo uma delas psicóloga (com doutoramento) e a outra neuropsicóloga (realizando o doutoramento). Estas PS vão também ser utilizadores finais do sistema.

Na primeira reunião com as PS foi realizada uma revisão de todos os requisitos funcionais antigos. Foram apresentados os requisitos através da interface do sistema antigo, permitindo o auxílio no levantamento de novos requisitos como também no refinamento dos antigos. Após esta primeira reunião com as PS os requisitos funcionais foram corrigidos e organizados consoante as funcionalidades principais do NexusBRaNT, sendo estes divididos nas seguintes categorias: requisitos funcionais gerais do sistema, requisitos funcionais da gestão de pacientes, requisitos funcionais da avaliação neuropsicológica e requisitos funcionais do programa treino (ponto 3 da Fig. 3.1).

Em seguida ocorreu outra reunião com as PS para a discussão de alguns requisitos funcionais que ficaram por esclarecer e descrição detalhada de outros, como por exemplo: definir quais os dados que seriam obrigatórios no registo de um novo paciente e quais os dados que deveriam ser apresentados na listagem de todos os pacientes (ponto 2 da Fig. 3.1).

Foram realizadas as modificações necessárias e finalizados os requisitos funcionais. Posto isto, os requisitos de tecnologia e não funcionais foram elicitados com informação recolhida anteriormente apresentada nas secções 2.3.3 e 2.3.4 como também com os restantes elementos do projeto BRaNT.

Ao longo do processo, várias versões intermédias foram apresentadas às PS. Com o objetivo de recolher *feedback* e solicitações simples para alterações em alguns requisitos funcionais. No decorrer das reuniões outros requisitos funcionais foram elicitados e refinados.

3.1.1. Evolução dos requisitos

Ao realizar a análise dos requisitos elicitados no sistema anterior detetou-se a existência de algumas lacunas. Os requisitos funcionais do sistema anterior não estavam categorizados, desta forma não era perceptível em que parte do sistema os requisitos estavam implementados. Nos novos requisitos funcionais elicitados (apresentados na secção) dividiu-se os mesmos por categorias, tendo como critério as funcionalidades principais da plataforma, sendo estes: requisitos gerais do sistema, requisitos da gestão de pacientes, requisitos da avaliação neuropsicológica e requisitos do programa de treino.

No sistema antigo existem requisitos que contêm informações redundantes, como por exemplo o requisito “O administrador do sistema deve ter a capacidade de criar contas de utilizador para os profissionais de saúde” e o requisito “O sistema deve permitir criar um registo de profissional de saúde com nome completo, email, data de adesão ao sistema, tipo de conta e senha de acesso”. Estes dois requisitos poderiam ser definidos como herança, um de criação de um novo utilizador e outro como sub-requisito com a descrição dos campos a serem inseridos, e não como dois requisitos distintos. Tem-se o exemplo do requisito elicitado RF2 (contém RF2.1 e RF2.1.1) onde toda a informação associada à criação de um novo utilizador está apenas nesse requisito. O mesmo ocorre nos requisitos antigos das atividades cognitivas de um programa de treino, sendo outro exemplo de qual os mesmos deveriam estar agrupados numa única categoria de requisitos e não como requisitos distintos.

Outro problema associado à redundância de requisitos foi a repetição do mesmo requisito, mas noutra perspetiva, como por exemplo “O sistema deve apenas deve permitir

o acesso por parte dos utilizadores autenticados” e “O sistema deve permitir ao utilizador iniciar sessão com o email e *password*”. Estes dois requisitos distintos referem-se ao mesmo processo, autenticação de um utilizador na plataforma com o email e a *password* que, por consequência, só se conseguem autenticar se tiverem registados na base de dados. Tendo em conta esta lacuna, resolveu-se esta redundância de requisitos reformulando e categorizando os novos requisitos descritos na secção 3.1.3.

É necessário também haver uma descrição detalhada dos dados referidos nos requisitos antigos, como por exemplo no requisito “O sistema deve permitir ao profissional de saúde registar novo paciente. O profissional de saúde deve inserir os dados sociodemográficos e clínicos do paciente”. Este requisito não indica quais os dados sociodemográficos e clínicos que deverão ser inseridos em detalhe, nem quais os campos que deverão ser obrigatórios de serem preenchidos pelo PS. Apesar das informações sociodemográficas e clínicas serem dados sobre o paciente, ambos são inseridos em formulários diferentes. Posto isto, os requisitos funcionais deveriam ser separados, tal como nos novos requisitos RF7, RF9 e seus sub-requisitos, onde as informações estão em requisitos distintos.

Através dos erros dos requisitos antigos, foi feita a análise, elicitación e refinação dos novos requisitos, obtendo assim uma nova lista de requisitos de modo a ter uma perceção mais concreta do sistema desenvolvido.

3.1.2. Requisitos de tecnologia

Os requisitos tecnológicos (RT) são os que caracterizam a base na qual o sistema deverá suportar, independentemente do utilizador [51]. Estes requisitos são:

RT1 - O *backend* do sistema deverá ser desenvolvido pela *framework* Express.js.

RT2 - O *frontend* do sistema deverá ser desenvolvido utilizando a biblioteca React.

RT3 - A base de dados deverá ser relacional.

RT3.1 - Deverá ser desenvolvida na ferramenta MySQL.

RT4 - O sistema, a API e a base de dados deverão estar *online*.

3.1.3. Requisitos funcionais

Os requisitos funcionais (RF) definem como é que deverá ser implementado o sistema de forma que os utilizadores consigam realizar as tarefas, representando assim o comportamento do sistema conforme as condições especificadas [51].

3.1.3.1. Requisitos funcionais gerais do sistema

RF1 - O sistema deverá garantir autenticação dos utilizadores, verificando sempre se os mesmos constam na base de dados do sistema.

RF1.1 - O sistema deverá permitir a inserção do email e da *password* para o utilizador autenticar-se.

RF1.2 - O sistema deverá notificar em caso de erro que o utilizador deverá autenticar novamente.

RF2 - O administrador do sistema poderá registar novos PS.

RF2.1 - O sistema deverá permitir registar um novo PS com o nome, email, data de adesão, tipo de conta (PS investigador ou PS clínico), *password*, célula profissional e contato telefónico.

- RF2.1.1 - O sistema deverá identificar como campos obrigatórios os dados: nome, email, data de adesão, *password*, célula profissional e contato telefónico.
- RF3 - O administrador do sistema poderá visualizar todos os utilizadores registados e apenas o ID, o nome e o email dos pacientes associados a cada utilizador.
- RF4 - O sistema deverá permitir ao administrador adicionar novos instrumentos de avaliação neuropsicológica.
- RF4.1 - O sistema deverá permitir registar o novo instrumento de avaliação com o nome, descrição, domínios e/ou subdomínios.
- RF4.1.1 - O sistema deverá identificar como campos obrigatórios todos os dados referidos no RF4.
- RF4.2 - O sistema deverá permitir adicionar os valores mínimos e máximos que cada subdomínio deverá conter.
- RF4.3 - O sistema deverá permitir edição de um instrumento de avaliação neuropsicológica.
- RF4.4 - O sistema deverá permitir remover um instrumento de avaliação neuropsicológica.
- RF5 - O sistema deverá permitir ao administrador associar e desassociar um paciente ao PS.
- RF6 - O sistema deverá permitir que os utilizadores terminem sessão.

3.1.3.2. Requisitos funcionais da gestão de pacientes

- RF7 - O sistema deverá permitir aos PS registar um novo paciente com o nome, email, morada, profissão, situação profissional, data de nascimento, sexo, escolaridade, lateralidade, agregado familiar, contato telefónico e estado civil.
- RF7.1 - O sistema deverá identificar todos os campos como obrigatórios.
- RF7.2 - O sistema deverá permitir aos PS visualizar os dados sociodemográficos do paciente.
- RF7.3 - O sistema deverá permitir aos PS atualizarem os dados sociodemográficos dos pacientes.
- RF7.4 - O sistema deverá permitir guardar informação sociodemográfica antiga do paciente.
- RF8 - O sistema deverá permitir aos PS visualizar os pacientes associados apenas a si.
- RF8.1 - O sistema deverá permitir filtrar/ordenar os pacientes pelo nome, idade e profissão.
- RF9 - O sistema deverá permitir aos PS visualizar a informação clínica do paciente.
- RF9.1 - O sistema deverá adicionar informação clínica do paciente com o tipo de informação (diagnóstico/história clínica, terapêutica farmacológica, exames complementares e registos clínicos das sessões), data de registo e descrição da informação.
- RF9.2 - O sistema deverá permitir eliminar informação clínica já registada no sistema.
- RF9.3 - O sistema deverá permitir aos PS editarem a informação clínica do paciente.
- RF9.4 - O sistema deverá permitir filtrar/ordenar a informação clínica pela data de registo, tipo de informação e descrição.
- RF9.5 - O sistema deverá permitir visualizar informação detalhada da informação clínica.
- RF10 - O sistema deverá permitir aos PS visualizar o perfil cognitivo dos seus pacientes.

- RF10.1 - O sistema deverá permitir calcular o perfil cognitivo do paciente através da última sessão de avaliação neuropsicológica.
- RF10.2 - O sistema deverá permitir calcular a interpretação clínica do paciente através da última sessão de avaliação neuropsicológica.
- RF11 - O sistema deverá apresentar as sessões de avaliações neuropsicológicas do paciente com o número de sessão da avaliação, a data, o PS e o instrumento de avaliação.
- RF11.1 - O sistema deverá permitir visualizar informação detalhada de cada avaliação neuropsicológica.
- RF11.2 - O sistema deverá permitir que o PS consiga editar a avaliação neuropsicológica (número de sessão, resultados, interpretação clínica e observações).
- RF11.3 - O sistema deverá permitir visualizar de forma gráfica as diferentes avaliações.
- RF11.4 - O sistema deverá permitir ao PS eliminar uma avaliação neuropsicológica desejada.
- RF11.5 - O sistema deverá permitir filtrar/ordenar as avaliações neuropsicológicas através do número de sessão, data, instrumento de avaliação e PS.
- RF12 - O sistema deverá apresentar os programas de treino do paciente.
- RF12.1 - O sistema deverá permitir aceder à informação detalhada dos programas de treino do paciente (número total de sessões, data da primeira sessão, número de vezes por semana, duração de cada sessão, desafio de adaptação e estado).
- RF12.2 - O sistema deverá permitir aos PS reagendar sessões de treino ainda não realizadas.
- RF12.3 - O sistema deverá permitir aos PS editar o programa de treino com os dados referidos no RF12.1.
- RF12.4 - O sistema deverá permitir filtrar/ordenar os programas de treino através do número de programa de treino, data de início, total de sessões, PS e estado.
- RF12.5 - O sistema deverá permitir ao PS eliminar o programa de treino desejado.

3.1.3.3. Requisitos funcionais da avaliação neuropsicológica

- RF13 - O sistema deverá permitir apresentar todos os instrumentos de avaliação.
- RF14 - O sistema deverá permitir criar uma sessão de avaliação neuropsicológica associado a um paciente.
- RF14.1 - O sistema deverá permitir adicionar mais do que um instrumento de avaliação na sessão de avaliação.
- RF14.2 - O PS poderá inserir os dados de acordo com os resultados obtidos pelo paciente em cada instrumento de avaliação selecionado.
- RF14.3 - O sistema deverá permitir a inserção da interpretação clínica nos instrumentos de avaliação quando contém.
- RF14.4 - Na existência de dados normativos no instrumento de avaliação, o sistema deverá permitir o PS visualizar as imagens correspondentes aos dados normativos do instrumento.

3.1.3.4. Requisitos funcionais do programa treino

- RF15 - O sistema deverá permitir criar um programa de treino associado a um paciente.

- RF15.1 - O sistema deverá permitir adicionar ao programa de treino os dados referidos no requisito RF12.1.
- RF15.1.1 - O sistema deverá identificar todos os campos como obrigatórios.
- RF15.2 - O sistema deverá permitir visualizar as tarefas de treino cognitivo recomendadas pelo sistema.
- RF15.3 - O sistema deverá permitir aos PS editarem os parâmetros das tarefas de treino cognitivo que pretendem que o paciente realize.
- RF15.4 - O sistema deverá permitir aos PS selecionar as tarefas de treino cognitivo que pretendem que o paciente realize, entre as que existem e as recomendadas.
- RF16 - O sistema deverá permitir aos PS visualizar todos os programas ativos dos pacientes associados.
- RF16.1 - O sistema deverá permitir aceder à informação detalhada de cada sessão de treino (número de sessão, data de início, data final, estado, parâmetros de intervenção e performance).
- RF16.2 - O sistema deverá permitir aos PS visualizarem as anotações em cada sessão de treino provenientes das atividades cognitivas realizadas pelo paciente.
- RF16.3 - O sistema deverá permitir filtrar/ordenar as sessões de treino através do número de sessão, data de início, data final e estado.
- RF17 - O sistema deverá permitir aos PS visualizar todos os programas finalizados dos pacientes associados.

3.1.4. Requisitos não funcionais

Os requisitos não funcionais (RNF) são difíceis de medir e de descrever. Estes referem-se ao sistema como um todo e não são restritos a um tipo de utilizador. Estes requisitos envolvem vários atributos de qualidade, tais como: *usability*, *security*, *safety*, *efficiency*, *portability*, entre outros [51].

3.1.4.1. Requisitos não funcionais de interface

- RNF1 - O sistema deverá ser simples e intuitivo.
- RNF2 - A interface deverá adaptar-se ao tamanho do ecrã, sendo responsiva.
- RNF3 - O sistema deverá suportar sistema externo de tradução de línguas.

3.1.4.2. Requisitos não funcionais de *software* e *hardware*

- RNF4 - O sistema deverá ser executado numa máquina com os seguintes recursos (1 core, com frequência 1,4 GHz e 512MB de RAM mínimo).
- RNF5 - O sistema deverá estar preparado para situações de *failover*.
- RNF6 - O sistema deverá funcionar, sem erros, nos vários sistemas operativos (por exemplo Windows, Linux e macOS).
- RNF7 - O sistema terá de estar acoplado a um servidor de armazenamento de dados.
- RNF8 - O servidor deverá notificar os *developers* quando ocorrer alguma falha no *site* ou na API.

3.1.4.3. Requisitos não funcionais de sistemas complementares

- RNF9 - Todos dados solicitados e enviados deverão ser armazenados de forma segura.
- RNF10 - O sistema deverá permitir uma utilização de 1000 utilizadores e 100 utilizadores em simultâneo.

3.1.4.4. Requisitos não funcionais de interoperabilidade

RNF11 - O sistema deverá permitir a implementação de novas rotas (solicitação e/ou envio de dados) para a integração com sistemas externos do projeto BRaNT.

RNF12 - O sistema deverá notificar o sistema externo de atividades cognitivas de erros ou atrasos de sessões.

3.1.5. Casos de utilização

Os casos de utilização (*use cases*) é uma técnica de modelação focada no que os utilizadores necessitam para alcançar o resultado que pretendem no sistema. Os diagramas de casos de utilização são constituídos por atores e por comportamentos, sendo estes a representação visual de alto nível das necessidades do utilizador. Cada ator poderá ter um ou mais casos de utilização associados, tendo em conta que a especificação destes dependem do contexto em questão. [51]. Em seguida listou-se casos de utilização identificados com os respetivos atores:

Pessoa: Engloba os atores PS e o Administrador.

UC1. Efetuar *login*.

Profissional de saúde (PS):

UC2. Registrar pacientes.

UC3. Consultar informação sociodemográfica e clínica dos pacientes associados.

UC4. Editar informação sociodemográfica e clínica.

UC5. Remover informação clínica.

UC6. Visualizar perfil cognitivo do paciente.

UC7. Criar uma sessão avaliação neuropsicológica.

UC8. Adicionar mais do que um instrumento de avaliação a uma sessão.

UC9. Adicionar a interpretação clínica na avaliação registada.

UC10. Consultar avaliações neuropsicológicas do paciente.

UC11. Editar avaliação neuropsicológica.

UC12. Remover avaliação neuropsicológica.

UC13. Criar um programa de treino cognitivo.

UC14. Consultar um programa de treino cognitivo do paciente.

UC15. Editar um programa de treino cognitivo.

UC16. Remover um programa de treino cognitivo.

Administrador:

UC17. Criar utilizadores.

UC18. Consultar informação dos utilizadores.

UC19. Editar informações dos utilizadores.

UC20. Associar/desassociar um paciente a um PS.

UC21. Adicionar novos instrumentos de avaliação.

UC22. Editar parâmetros do instrumento de avaliação

UC23. Remover parâmetros de avaliação.

Developer:

UC24. Verificação e correção de erros recebidos.

UC25. Dar suporte aos utilizadores.

Após a listagem dos casos de utilização desenhou-se o diagrama de casos de utilização referentes ao funcionamento geral do NexusBRaNT, sendo este apresentado na Fig. 3.2.

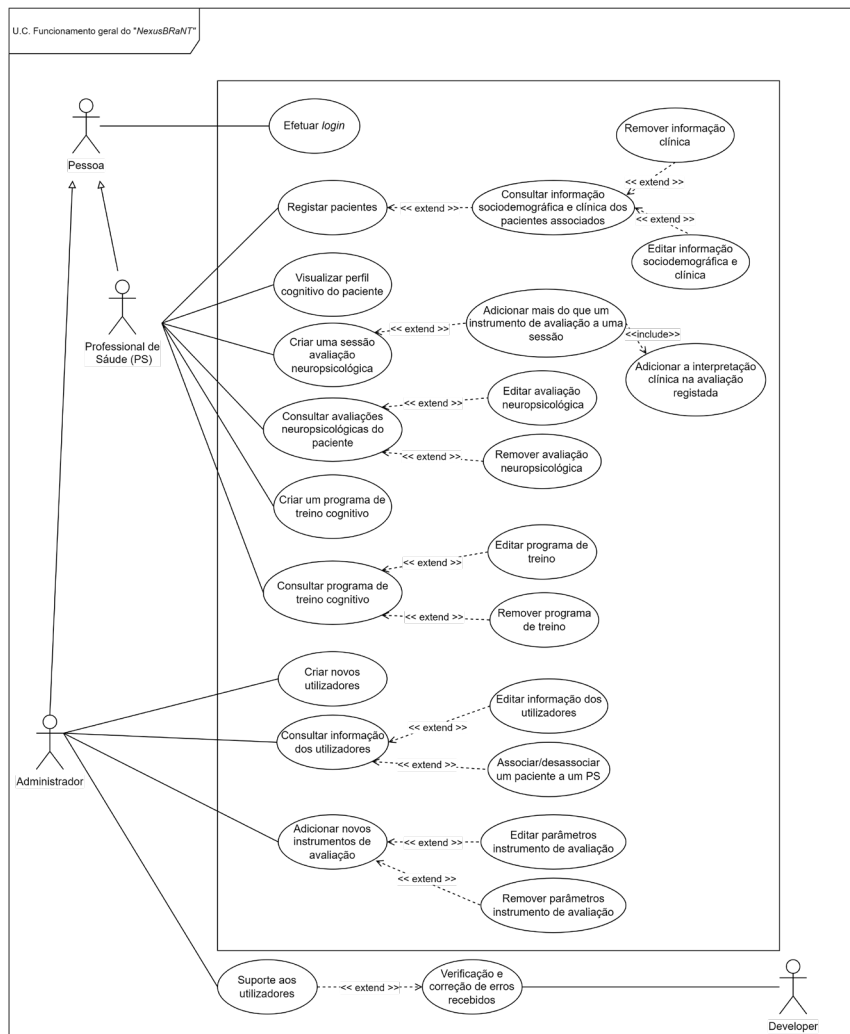


Fig. 3.2 – Diagrama de Casos de Utilização: Funcionamento Geral do NexusBRaNT.

Observando a Fig. 3.2 é possível identificar os diversos atores, tais como: “Pessoa”, “Profissional de saúde”, “Administrador”, “Developer” e “Servidor”. Os atores “Profissional de saúde” e “Administrador” são subtipos de “Pessoa”, herdando assim os seus casos de utilização.

Os dois atores filhos de “Pessoa” efetuam o *login*. O “Profissional de saúde” é o ator principal do sistema e está envolvido em diversas atividades, tais como: registar, consultar e editar informação sociodemográfica e clínica dos pacientes associados, visualizar perfil cognitivo do paciente, criação de uma sessão de avaliação neuropsicológica e associar mais do que um instrumento de avaliação a uma sessão, consultar e editar avaliações neuropsicológicas, criação, consulta, edição e remoção de programas de treinos cognitivos. O ator “Administrador” tem como atividades associadas aquelas que permitem a administração e gestão do sistema, tais como: criação, consulta, edição e remoção de novos utilizadores (PS), associar um paciente a um PS, criação, consulta, edição e

remoção novos de instrumentos de avaliação, adicionar tarefas cognitivas e dar suporte aos utilizadores.

O último ator do sistema é o “*Developer*” que faz a verificação e correção de erros reportados pelos outros utilizadores do sistema, dando suporte aos mesmos. No diagrama estão apresentados os casos de utilização mais relevantes e gerais do sistema. Através dos requisitos funcionais (secção 3.1.3) é possível verificar que estes atores têm outros comportamentos além dos que foram apresentados no diagrama de casos de utilização.

3.2. Arquitetura proposta

A arquitetura proposta do sistema contém 4 camadas distintas, sendo estas: *presentation layer*, *logic layer*, *data layer* e *common layer*. Existem dois tipos de utilizadores no sistema: os PS e o administrador no NexusBRaNT. Na Fig. 3.3 está apresentada a arquitetura proposta.

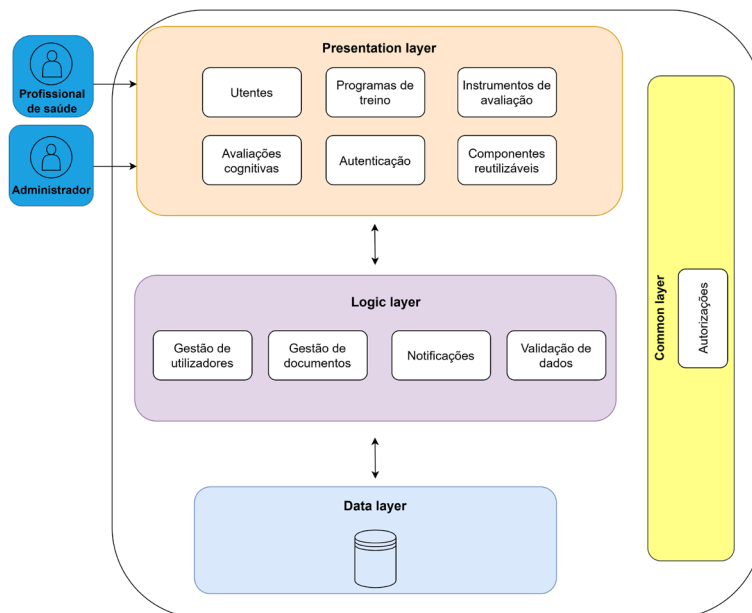


Fig. 3.3 – Arquitetura proposta do NexusBRaNT com 4 camadas: *presentation layer* (interação entre o utilizador e o sistema), *logic layer* (manipulação da informação e solicitação da informação na camada inferior), *data layer* (armazenamento dos dados) e *common layer* (autorizações em comum nas camadas).

A arquitetura proposta segue o padrão arquitetural MVC, sendo que este padrão consiste em 3 tipos de componentes: o modelo, a vista e o controlador, com funcionalidades distintas, mas que comunicam entre si. O modelo contém todos os dados do sistema. A vista apresenta os dados necessários ao utilizador e interage com o mesmo. E o controlador realiza a intermediação entre os dois componentes anteriores e permite a gestão de mudanças de estado através de notificações. De uma forma geral o MVC tem como vantagens a facilidade no desenvolvimento, a realização de testes em paralelo, pouco impacto nos outros componentes quando é necessário modificar um e facilidade em atualizar o sistema [52].

A *presentation layer* no MVC será a vista, sendo responsável pela interação entre o utilizador e o sistema. Esta camada contém algumas vistas onde o PS e o administrador

poderão interagir com o sistema, sendo estas: pacientes, programas de treinos, instrumentos de avaliação, avaliações neuropsicológicas, autenticação e componentes reutilizáveis.

O componente controlador na arquitetura proposta é a *logic layer* que manipula a informação consoante o necessário, solicita informação à *data layer* e apresenta os resultados à *presentation layer*. Na camada *logic layer* é possível a gestão de documentos, a gestão de utilizadores, notificações e validação de dados.

Por fim o componente modelo presente na *data layer* permite aceder aos dados armazenados e dar resposta às solicitações feitas pela camada superior. Esta camada contém a base de dados, uma vez que os modelos não são diretamente acessíveis pela camada superior.

A *common layer* contém funcionalidades comuns às diversas camadas existentes. Nesta arquitetura, a *common layer* apenas inclui as autorizações que serão necessárias no sistema. Outras configurações gerais serão posteriormente adicionadas nesta camada.

Ao longo do processo de desenvolvimento do NexusBRaNT esta arquitetura proposta poderá ser adaptada, consoante as alterações sugeridas.

3.3. Método de desenvolvimento

O método de desenvolvimento definido para a implementação do NexusBRaNT foi o método iterativo para um desenvolvimento incremental. Os requisitos foram priorizados de forma incremental, de acordo com os pedidos das especialistas que ajudaram na elicitação dos requisitos. A cada ciclo do processo de desenvolvimento, após a implementação dos requisitos priorizados, era feita novamente uma priorização dos requisitos a serem desenvolvidos no próximo *sprint*. Ao longo do processo, conforme as necessidades das especialistas foram elicitados e priorizados novos requisitos, por exemplo a notificação quando a sessão de treino está atrasada.

Começou-se por desenvolver os requisitos que envolviam o registo e autenticação de utilizadores (pois estes eram requisitos cruciais a nível funcional e arquitetural), registo e visualização dos pacientes e em seguida a implementação dos requisitos funcionais da gestão de pacientes. Implementou-se a visualização e edição da informação sociodemográfica e clínica do paciente, registo, visualização e edição das avaliações neuropsicológicas. Posto isto, a implementação da visualização e registo dos programas de treinos cognitivos foram os requisitos desenvolvidos. A visualização do perfil do profissional de saúde e edição das suas informações foram desenvolvidas paralelamente aos outros requisitos.

Durante o processo de desenvolvimento, e paralelamente à implementação de cada funcionalidade, foram realizados testes de usabilidade, de integração com a API e de performance. A cada 4 requisitos desenvolvidos, as funcionalidades eram apresentadas as especialistas, aos membros do projeto e orientadores, de forma a obter a validação funcional e técnica de cada. Conforme o *feedback* recebido foram realizadas modificações ou aprovação da funcionalidade.

4. Implementação

Neste capítulo é apresentada toda a implementação do sistema, as ferramentas/tecnologias utilizadas para auxiliarem o desenvolvimento do mesmo e toda a evolução do sistema (base de dados, *backend* e *frontend*). Devido às lacunas na plataforma anteriormente desenvolvida (secção 2.2.3), optou-se pela implementação do NexusBRaNT como novo projeto, sem dependências do anterior.

4.1. Ferramentas de desenvolvimento

Ao longo do projeto foram utilizadas várias ferramentas de *software* que auxiliaram no desenvolvimento do sistema. Serão apresentadas as ferramentas Visual Studio Code, GitHub, MySQL Workbench, Wampserver64, PhpMyAdmin, Postman e a biblioteca Material – UI, que contribuíram em diferentes partes do projeto. As funcionalidades de cada ferramenta são as seguintes:

- Visual Studio Code²: o editor de código-fonte utilizado para a implementação e edição do código do sistema desenvolvido;
- GitHub³: utilizado para a hospedagem do código desenvolvido, controlo de versões e colaboração entre membros;
- MySQL Workbench⁴: *design*, criação e manutenção para base de dados relacional desenvolvida e modificada;
- Wampserver64⁵: utilizado para ter um servidor local e acesso ao PhpMyAdmin;
- PhpMyAdmin⁶: gerir a base de dados pela *web* e visualização em tempo real dos dados a serem inseridos/modificados/removidos pelo sistema;
- Postman⁷: utilizada para usar a API que foi desenvolvida e visualização simplificada dos testes nas rotas criadas;
- Material – UI⁸: biblioteca utilizada para o *design* da interface do sistema, que contém componentes React de código aberto.

Utilização destas ferramentas auxiliaram na projeção e no desenvolvimento ágil do sistema que foi implementado.

4.2. Base de dados

A base de dados utilizada no NexusBRaNT já tinha sido modelada parcialmente (como referido na secção 1.3) e a mesma contém todas as informações do projeto de investigação BRaNT. Tendo em conta os requisitos revistos e elicitados, a base de dados sofreu alterações na estrutura, tanto em adição e remoção de tabelas como na modificação, adição e remoção de atributos nas tabelas. Estas alterações tiveram sempre em conta os requisitos elicitados e os novos requisitos que foram elicitados e refinados durante o desenvolvimento do sistema.

²Visual Studio Code, <https://code.visualstudio.com/>

³GitHub, <https://github.com/>

⁴MySQL Workbench, <https://www.mysql.com/products/workbench/>

⁵Wampserver64, <https://www.wampserver.com/en/>

⁶PhpMyAdmin, <https://www.phpmyadmin.net/>

⁷Postman, <https://www.postman.com/>

⁸Material – UI, <https://mui.com/pt/material-ui/>

Em seguida são apresentados alguns exemplos de alterações na base de dados:

- Adição de tabelas:
 - Devido ao requisito RF14 foi necessário adicionar a tabela “*Sessions_Assessment*” que contém o ID da sessão de avaliação e a data respetiva. Desta forma, é possível associar à sessão a avaliação neuropsicológica realizada. Na Fig. A.2 (anexo A) estão apresentadas ambas as tabelas e a sua relação;
 - No requisito RF7 é solicitado o estado civil do paciente, sendo este um novo pedido na informação sociodemográfica. Assim sendo, foi necessária a criação de uma tabela “*MaritalStatus*” que contém todos os estados civis conhecidos. Na Fig. A.1 (anexo A) é possível observar a verificação da relação da informação sociodemográfica com o estado civil. Tal como o estado civil, também foi necessário adicionar uma tabela para a situação profissional do paciente, sendo esta “*ProfessionalSituation*”. Na Fig. A.3 no anexo A está apresentada a tabela e sua relação com a tabela da informação sociodemográfica;
 - Ao longo do desenvolvimento foram sugerindo novos requisitos, um deles foi o requisito RF14.4. Como consequência, foi necessário adicionar novas tabelas na base de dados. Dividiu-se a interpretação clínica em duas categorias: cognitiva e não cognitiva e criou-se as tabelas “*NormativeDataCognitive*” e “*CutoffScore*”. A interpretação clínica cognitiva envolve média e desvio-padrão ou pontuação escalar, sendo que desta forma criou-se as tabelas que contém a descrição de ambos os tipos de interpretação (“*PatternDeviationInterpertation*” e “*ScalarScoreInterpertation*”). Como referido no requisito RF14.4 também foi solicitado a inserção da média e o desvio padrão ou a pontuação escalar que o paciente se enquadrada, criando assim a tabela “*PatternDeviationScalarScore*”. Na Fig. A.4 do anexo A, está apresentada as relações e tabelas criadas. Também tem-se os pontos de corte (tabela “*CutoffScore*” criada) e a mesma necessita da interpretação clínica não cognitiva. Então criou-se a tabela “*NoCognitiveInterpertation*”, que contém todos os dados necessários para a interpretação clínica de um determinado instrumento de avaliação. As relações destas tabelas anteriormente referidas estão apresentadas na Fig. A.5.
 - Devido ao requisito RNF12 foi necessário adicionar a tabela “*AlarmNotification*” que será utilizada no futuro para armazenar notificações dos programas de treinos que provém do sistema externo desenvolvido por outros elementos do projeto de investigação. Na Fig. A.6 do anexo A está apresentada a tabela criada e a sua relação com a tabela “*TrainingPrograms*”.
- Modificação, adição e remoção de atributos:
 - Observando a Fig. A.7 a) e b) (anexo A) é possível visualizar a diferença entre a tabela original (Fig. A.7 a)) e a tabela atual (Fig. A.7 b)) dos programas de treinos. Tendo em conta o requisito RF15, os atributos da tabela “*Training_Programs*” foram alterados, por exemplo o tipo, o nome e propriedades. Alguns atributos foram adicionados consoante a refinação do requisito.
 - A tabela das sessões do treino cognitivo também teve diversas modificações devido a refinação do requisito RF15.1, principalmente no aumento dos atributos da mesma e modificação do seu nome, de forma a distinguir as sessões dos programas de treino e as sessões das avaliações

neuropsicológicas. Na figura Fig. A.8 a) e b) estão apresentadas a tabela antiga e atual, respetivamente;

- As tabelas dos instrumentos de avaliação também já existiam na base de dados, mas foram revistas e modificadas consoante o necessário. Por exemplo, a tabela “*ToulousePieronCancellationTest*”, apresentada na Fig. A.9 a) e b), foi transformada e corrigida. Outro exemplo foi a tabela “*PhonemicSemanticVerbalFluence*” que foi transformada, sendo que os atributos foram corrigidos e outros adicionados. Na Fig. A.10 a) b) estão apresentadas tabelas antigas e atuais, respetivamente. Todos os atributos alterados referentes aos instrumentos de avaliação foram consultados com as PS e os mesmos foram descritos em tabelas de Excel para facilitar a documentação e manutenção futura. Nas Tabela A.1 a Tabela A.20 do anexo A estão apresentadas as descrições dos campos dos instrumentos de avaliação facultadas pelas PS que elicitaram os requisitos e a organização para as modificações realizadas às tabelas da base de dados relacionadas com os instrumentos de avaliação.
- Alterações das inserções:
 - Existem tabelas na base de dados que contêm já dados inseridos. Um dos exemplos é a tabela “*NPAElementDescriptor*” que contém a descrição de todos os campos de cada instrumento de avaliação. Os dados já inseridos anteriormente foram corrigidos consoante as informações cedidas pelas PS, referidas no exemplo anterior.
 - Algumas tabelas como “*Relationship*” e “*Clinical_Info_Types*” também tiveram mais inserções de dados. A “*Relationship*” é o tipo de relação que o paciente tem com o membro da família (por exemplo, pai, mãe, filho, entre outros). A tabela “*Clinical_Info_Types*” são os tipos de informação clínica que será posteriormente inserida no sistema (por exemplo, diagnóstico principal, exame complementares, entre outros).
 - Em algumas das novas tabelas, como por exemplo “*ProfessionalSituation*”, “*Activities*”, “*NoCognitiveInterpretation*”, “*PatternDeviationInterpretation*” e “*ScalarScoreInterpretation*”, são inseridos manualmente dados. Estes dados são consultados para visualização de algumas informações ou campos de seleção.

4.2.1. Visão geral

Ao longo do desenvolvimento do sistema a base de dados foi regularmente modificada. No anexo A, a Fig. A.11 apresenta a base de dados inicial (parcialmente modelada). Em seguida, no mesmo anexo A, na Fig. A.12 está apresentada a base de dados final do projeto.

Observando a Fig. A.12 é possível visualizar secções ou agrupamentos distintos na base de dados, sendo que estas correspondem a diferentes partes lógicas do projeto. Nem todas as tabelas apresentadas na base de dados são utilizadas no NexusBRaNT, pois a base de dados engloba todo o projeto BRaNT e algumas tabelas serão apenas utilizadas em trabalhos futuros. Estes agrupamentos podem ser divididos em quatro grupos principais:

- Avaliações neuropsicológicas:
 - Nesta secção estão todas as tabelas (apresentadas no retângulo verde) que correspondem às avaliações neuropsicológicas, tais como: os instrumentos de

avaliação, a descrição dos campos dos instrumentos de avaliação, os dados normativos dos instrumentos de avaliação, o perfil cognitivo, a interpretação clínica e a tabela principal (em rosa) que contém todas as relações necessárias para uma avaliação neuropsicológica.

- Por exemplo, a tabela “*AssessmentByInstrument*” (em rosa) contém as relações entre os instrumentos de avaliação, ou seja, o paciente, o PS e a sessão de avaliação (referido anteriormente na secção 2.3.4).
- Instrumentos de avaliação:
 - Todos os instrumentos de avaliação neuropsicológica estão apresentados nas tabelas cor de laranja;
 - Cada tabela contém os atributos correspondentes a cada instrumento;
 - A tabela “*NPA*” contém todos os instrumentos e esta relaciona-se com a tabela “*NPAElementDescriptor*” (referida anteriormente na secção 4.2).
- Informações do paciente e relações com os PS:
 - No retângulo azul estão apresentadas todas as tabelas que correspondem às informações dos pacientes, tais como: a informação pessoal, a sociodemográfica e a clínica;
 - Contém também as tabelas com a informação dos PS e relação entre o paciente e o PS.
- Programas de treinos e tarefas cognitivas:
 - Toda a informação acerca dos programas de treino cognitivo e atividades cognitivas estão apresentadas nas tabelas do retângulo roxo;
 - As tabelas contém informações dos programas de treino, das sessões de treino, das informações essenciais para as atividades cognitivas e dos resultados provenientes das atividades realizadas pelos pacientes.

O ficheiro do diagrama apresenta a base de dados de uma forma geral. As Tabela A.21 a Tabela A.72 do anexo A mostram em detalhe as várias tabelas da base de dados utilizadas no desenvolvimento do projeto.

4.2.2. Dificuldades

Devido ao facto de iniciar-se este projeto com uma base de dados parcialmente desenvolvida, foi necessário fazer uma análise prévia da mesma para realizar a implementação do sistema. Para tal, realizaram-se algumas reuniões para obter uma visão geral da mesma, análise e esclarecimento de dúvidas. Ao longo do desenvolvimento do sistema consultou-se as tabelas de modo a conseguir relacionar as informações e, desta forma, apresentar os dados solicitados ou inserir os dados enviados.

Consoante a utilização da base de dados existiram relações que se alteraram, como por exemplo, relações que deveriam ser muitos para muitos e eram um para muitos. Algumas das tabelas existentes na base de dados continham redundância nas informações, por isso sempre que era adicionada uma nova tabela teve-se em consideração não repetir informações e eliminar as informações repetidas.

Ao rever-se todas as informações dos instrumentos de avaliação teve-se de corrigir o tipo de cada atributo nas tabelas dos instrumentos. Existem instrumentos de avaliação que apenas contém campos com números inteiros e outros com casas decimais (uma ou duas

casas decimais). Desta forma, corrigiu-se todas as tabelas e alterou-se o tipo de campo dos instrumentos de avaliação.

Durante o desenvolvimento foi necessário adicionar novos instrumentos de avaliação e, conseqüentemente, criaram-se tabelas para os mesmos. Devido à complexidade das relações não bastou apenas adicionar as tabelas na base de dados, uma vez que foi também necessário verificar as relações e adicionar as informações acerca das mesmas. Tem-se o exemplo das tabelas “NPA”, onde foram adicionados os novos instrumentos de avaliação e a tabela “NPAElementDescriptor”, em que foi necessário adicionar todas as informações dos campos dos instrumentos novos.

Existiam tabelas e campos que não continham descrições/observações ou algumas das que continham não existia uma correta correspondência. Para tal, foi necessário prosseguir na correção dos mesmos e adicionar os que faltavam, tornando assim a base de dados mais perceptível, caso seja necessária manutenção no futuro.

De forma geral, com o desenvolvimento do sistema e as conseqüentes consultas à base de dados, tornou-se mais fácil a utilização e manutenção da mesma.

4.3. Backend

O *backend* no projeto desenvolvido teve como principal objetivo a gestão dos dados dos pacientes, das avaliações neuropsicológicas, dos programas de treino e dos utilizadores base de dados através da implementação de uma API em ExpressJs (referida anteriormente na secção 2.3.4). A API desenvolvida dá suporte ao *frontend* que será apresentado posteriormente e também aos sistemas externos ao NexusBRaNT, como por exemplo as atividades cognitivas que foram desenvolvidas por outros colegas.

Na arquitetura proposta do sistema (referida na secção 3.2) segue o padrão arquitetural MVC, sendo que o “*Model*” e o “*Controller*” estão implementados no lado da API e a “*View*” no *frontend*.

Na Fig. 4.1 está apresentada a organização da API.

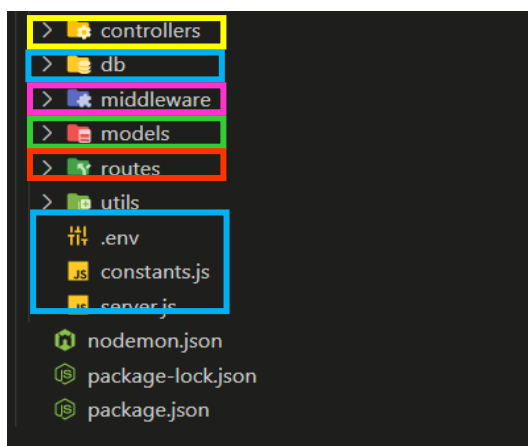


Fig. 4.1 – Organização dos ficheiros na API.

É possível observar na Fig. 4.1 que existem 5 grupos principais: as configurações (azul), os modelos (verde), os controladores (amarelo), os *middlewares* e validadores (rosa) e as rotas (laranja). Em seguida serão elucidados cada grupo.

4.3.1. Configurações

As configurações principais (Fig. 4.1 a azul) realizadas na API dividem-se em três secções: a ligação à base de dados, as definições principais das rotas no servidor e a definição de constantes e variáveis de ambiente.

4.3.1.1. Ligação à base de dados

Utilizou-se a biblioteca *mysql2*⁹ para realizar a ligação à base de dados, sendo esta integrada no gestor de pacotes *npm*¹⁰. Criou-se uma classe chamada “*DBConnection*” que contém o construtor e 3 funcionalidades: a verificação da conexão à base de dados, a realização de uma *query* e a realização de uma transação.

No construtor são criados 2 tipos de conexões, através da função “*createPool*”, sendo uma um pedido simples e outra para uma transação (referido posteriormente). Deste modo em ambos são designados a localização, as credenciais e os dados essenciais para o acesso à base de dados (*host*, *user*, *password* e *port*), através do *PhpMyAdmin*. Por questões de segurança o utilizador que acede à base de dados através do *PhpMyAdmin* tem permissões deduzidas.

A função “*checkConnection*” certifica a conexão à base de dados. Esta verifica se ocorreu algum erro na conexão à base de dados enviando uma mensagem para a consola da API, deste modo, identificando o problema ou o sucesso da conexão.

Para a execução de uma *query* à base de dados, foi implementada a funcionalidade assíncrona “*query*” apresentada na Fig. 4.2, que recebe dois parâmetros: “*sql*” e “*values*”, sendo estes uma *string* que contém a estrutura da *query* que pretende ser executada e os valores necessários para a complementação da mesma, respetivamente. É retornada uma função “*new Promise*” (objeto utilizado para processos assíncronos) que tem como parâmetros “*resolve*” e “*reject*”, é executado o objeto “*pool*” (criado no construtor), enviando o “*sql*”, os “*values*” e a função “*callback*” como parâmetros. A função *arrow* “*callback*” tem como parâmetros “*error*” e “*result*”. Em caso de erro, realiza “*reject*” no “*error*” e retorna o mesmo. Quando não ocorrem erros é realizado “*resolve*” com o “*result*”. Em caso de erro, na função “*Promise*”, ainda é realizado um *catch* que obtém “*HttpStatusCodes*” e adiciona essa informação ao erro, retornando-o.

⁹ **mysql2**, <https://www.npmjs.com/package/mysql2>

¹⁰ **npm**, <https://www.npmjs.com/>

```

async query(sql, values) {
  return new Promise((resolve, reject) => {
    const callback = (error, result) => {
      if (error) {
        reject(error);
        return error;
      }
      resolve(result);
    };

    // execute will internally call prepare and query
    this.pool.execute(sql, values, callback);
  }).catch(err => {
    const mysqlErrorList = Object.keys(HttpStatusCodes);
    // convert mysql errors which in the mysqlErrorList list to http status code
    err.status = mysqlErrorList.includes(err.code) ? HttpStatusCodes[err.code] : err.status;
    return err;
  });
}

```

Fig. 4.2 – Função “query”.

A função assíncrona que executa as transações chama-se “*transaction*” (apresentada na Fig. 4.3) que tem como parâmetros a “*pool*” e “*callback*”. É realizado no “*pool*” a função “*getConnection()*” e, em seguida, a função “*beginTransaction()*”. A mesma função contém um “*try...catch*”, sendo que no bloco “*try*” é retornado um objeto *json* com informação sobre o sucesso da transação. No bloco “*catch*”, que recebe como parâmetro o “*error*” que, quando detetado, retorna um objeto *json* com informação do insucesso, do erro e o estado do mesmo.

```

async transaction(pool, callback) {
  const conn = await pool.getConnection();
  await conn.beginTransaction();
  let successTrans = false;
  let data = {};
  let status = {};
  try {
    await callback(conn);
    await conn.commit();
    conn.release();
    successTrans = true;
    data = "Success";
    return { success: successTrans, data: data };
  } catch (error) {
    await conn.rollback();
    conn.release();
    successTrans = false;
    data = error;
    const mysqlErrorList = Object.keys(HttpStatusCodes);
    status = mysqlErrorList.includes(error.code) ? HttpStatusCodes[error.code] : error.status;
    return { success: successTrans, data: data, status: status };
  }
}

```

Fig. 4.3 – Função “*transaction*”.

4.3.1.2. Servidor

No ficheiro “*server.js*”, apresentado na Fig. 4.1, estão definidas as configurações para executar a API no servidor. Neste ficheiro as principais funções são: as inicializações do *express* e do ambiente através da biblioteca *dotenv*¹¹, a definição da porta no qual o processo *env* irá executar (neste caso na porta 3333), a definição dos endereços iniciais das rotas e a criação do servidor HTTPS.

¹¹ **dotenv**, <https://www.npmjs.com/package/dotenv>

Na Fig. 4.4 estão apresentados os endereços iniciais das rotas.

```
app.use(`/api/users`, userRouter);
app.use(`/api/patients`, patientRouter);
app.use(`/api/clinical-info`, cliInfoRouter);
app.use(`/api/assessments`, userAssRouter);
app.use(`/api/form`, formsRouter);
app.use(`/api/training-program`, trainRouter);
```

Fig. 4.4 – Definição inicial do endereço das rotas.

É possível observar na Fig. 4.4 que utilizou-se a função “use” que recebe como parâmetros o endereço inicial da rota e o módulo *router* correspondente ao endereço. De modo geral, dividiu-se as rotas da seguinte forma, sendo estas:

- /api/users – Rotas dos utilizadores (PS);
- /api/patients – Rotas dos pacientes;
- /api/clinical-info – Rotas das informações clínicas dos pacientes;
- /api/assessments – Rotas das avaliações neuropsicológicas dos pacientes;
- /api/form – Rotas das informações necessárias para os formulários no *frontend*;
- /api/training-program – Rotas dos programas treinos cognitivos dos pacientes.

Quando uma rota não é encontrada é apresentada uma mensagem na consola da API com o erro. Para a criação do servidor HTTPS são definidos a *key* e o certificado necessários para o protocolo de segurança SSL (*Secure Sockets Layer*).

4.3.1.3. Definição de constantes e variáveis de ambiente

Na Fig. 4.1 é possível verificar dois ficheiros, o “*constansts.js*” e “.*env*”. O primeiro ficheiro corresponde a *class constants* onde são declaradas as constantes com os nomes de todas as tabelas da base de dados. Optou-se por definir em constantes e num único ficheiro pelo facto de que ao longo da API são utilizadas diversas vezes os nomes das tabelas em ficheiros distintos. Desta forma, ao colocar num único local, caso existam alterações no nome da tabela na base de dados é apenas necessário mudar o nome da tabela neste ficheiro e evitando também que a escrita do nome das tabelas, ao longo da API, seja introduzido errado. Deste modo, reduziu-se assim o tempo de pesquisa onde as tabelas estão sendo utilizadas, em caso de alteração do nome das mesmas, e evitou-se redundância de código, pois em alguns ficheiros definia-se o nome da tabela como uma constante.

No segundo ficheiro, estão declaradas as variáveis de ambiente, neste caso as configurações locais em tempo de execução (porta e *secret jwt (JSON Web Token)*).

4.3.2. Visão geral

Na Fig. 4.5 é apresentado o funcionamento geral da API.

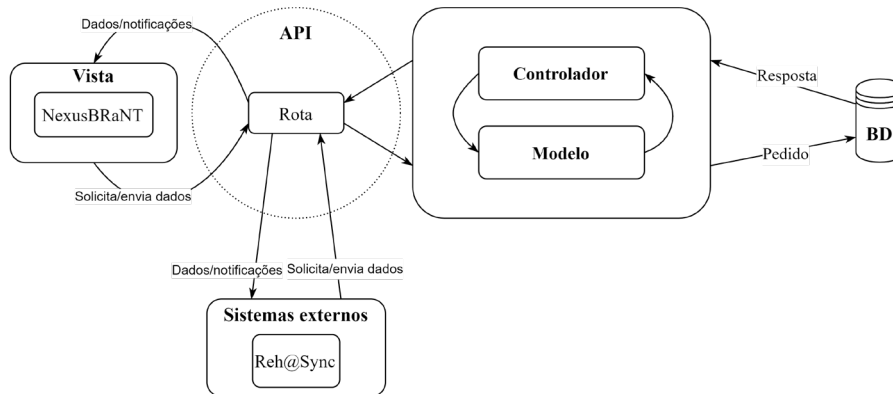


Fig. 4.5 – Funcionamento geral da API.

Primeiramente o *site* NexusBRaNT solicita e/ou envia dados para API através da rota. Posteriormente será apresentado de forma geral como é feita a comunicação entre o *frontend* e a API.

É possível enquadrar a Fig. 4.5 na Fig. 3.3, pois a *presentation layer* contém a vista, a API é a ligação entre a *presentation layer* e a *logical layer*, o controlador está contido na *logical layer* e o modelo e a base dados estão apresentados na *data layer*. A rota no lado da API executa a função do controlador definida na mesma. A função no controlador executa e aguarda resposta da função do modelo definida antes de retornar para a rota novamente. A função no modelo executa o pedido à base de dados, a sua resposta é formatada e retornada ao controlador, à rota e, por fim, os dados e/ou as notificações são apresentadas na *view*.

A figura abaixo (Fig. 4.6) apresenta um exemplo do funcionamento da API quando é inserido uma informação clínica de um paciente.



Fig. 4.6 – Funcionamento da rota de registro da informação clínica do paciente.

É possível observar na Fig. 4.6 que a rota contém o endereço (/create/:id), o validador dos dados (*createClinicalInfoSchema*) e a função “*awaitHandlerFactory()*” que tem como parâmetro a função (*create*) do controlador *ClinicalInformationController*. No controlador tem-se a função “*create*” que inicialmente executa a função “*checkValidation*” (verificação dos dados no validador correspondente), a definição do objeto *json params* e execução da função “*create*” do modelo. A função “*create*” no modelo “*ClinicalInformation*” executa a *query* de inserção dos dados da informação clínica na base de dados e retorna o resultado para o controlador. Em caso de falha na inserção dos dados é enviada uma exceção com o erro. Caso contrário, é enviada uma resposta informando que a informação foi adicionada com sucesso.

Este tipo de procedimento é realizado no decorrer da API. Nos subcapítulos seguintes serão apresentados especificamente as rotas, os controladores, os modelos e os *middlewares* e os validadores.

4.3.3. Rotas

Existem diferentes tipos de rotas ao longo da API: *get* (solicitar dados), *post* (enviar dados e criar um recurso), *delete* (eliminar dados) e *put* (atualizar dados).

Na Fig. 4.7 está apresentada a organização das rotas na API.

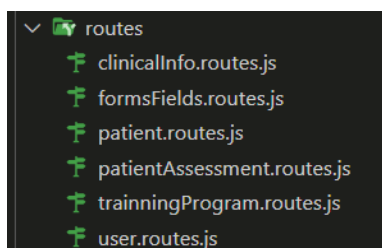


Fig. 4.7 – Organização das rotas na API.

Cada ficheiro *JavaScript* apresentado na Fig. 4.7 corresponde a um módulo que contém as rotas necessárias para cada grupo, dividindo-se em 6 grupos, como referido na secção 4.3.1.2. Todos os ficheiros apresentados contêm definições em comum, tais como: a importação do módulo *express*, a criação do *express router*, a importação do *middleware* “*awaitHandlerFactory*”, a solicitação (*get/post/delete/put*) ao *router* criado e exportação do módulo.

O *middleware* “*awaitHandlerFactory*” contém a função “*awaitHandlerFactory*” que recebe como parâmetro o *middleware* e retorna uma função assíncrona com os parâmetros “*req*”, “*res*” e “*next*”. Esta função contém um “*try...catch*”, sendo que no bloco *try* é aguardada a resposta da função *middleware(req,res,next)*. Caso seja detetado um erro, o bloco *catch* é ativado e executa-se “*next(err)*”.

As rotas dos utilizadores correspondem ao ficheiro “*user.routes.js*” (apresentado por último na Fig. 4.7). Quase todas as funções neste ficheiro acedidos nas rotas são do controlador “*UserController*”, sendo as restantes do “*PatientController*”, estando estas assinaladas na descrição. Este ficheiro contém as seguintes rotas:

- Rota do login do utilizador: é realizado as verificações se o utilizador existe e se *password* corresponde à existente na base de dados e é enviado uma mensagem de sucesso/erro para realizar *login* no *site*. A função acedida na rota é o “*login*”.

- Rota do registo do utilizador: esta rota contém o validador que verifica os dados do registo, antes da inserção na base de dados, e acede à função “*create*” para a inserção de um utilizador.
- Rota para obter todos os utilizadores: para obtenção de todos os PS é utilizado um *get*. A função acedida na rota é “*getAllUsers*”. Esta rota é utilizada apenas quando o utilizador autenticado é o administrador, sendo que esta devolve apenas os PS que não são administradores.
- Rota para obter um utilizador através do seu ID: para obter a informação de apenas um profissional de saúde é utilizada esta rota. É necessário enviar no endereço da rota o ID do utilizador que está a ser solicitado as informações.
- Rota para atualizar utilizador: tal como a rota do registo, é essencial um validador para verificar os dados alterados antes de serem realizadas as atualizações na base de dados. O validador é o mesmo que o do registo, pois os dados inseridos podem ser todos editados. A função “*updateInfoUser*” é acedida na rota.
- Rota para obter todos os pacientes que não estão associados: para obter os pacientes que não estão associados a nenhum PS é utilizada esta rota. A função acedida no controlador “*PatientController*” é a “*showNotAssociates*”.
- Rota para associar o paciente: é criada uma inserção na tabela que associa o paciente a um PS. Deste modo, esta rota permite associar um paciente, que não contém nenhuma relação, a um PS selecionado. Posteriormente ao sucesso desta rota, o PS ao autenticar-se já contém toda a informação do novo paciente associado a si. A função acedida na rota é “*createAssociatePatient*” do “*PatientController*”.
- Rota de desassociar o paciente: é eliminada a relação do paciente e do PS na tabela das relações entre ambos. Nenhum dado no paciente é eliminado, nem do PS. É acedida à função “*deletePatient*” do controlador “*PatientController*”.

As rotas dos pacientes dividem-se em 4 grupos principais, sendo estes: pacientes, informações clínicas, avaliações neuropsicológicas e programas de treino. O ficheiro “*patient.routes.js*” contém as rotas associadas às informações sociodemográficas, à edição dos dados, ao perfil cognitivo, às informações gerais das avaliações neuropsicológicas, entre outros. As rotas do ficheiro “*patient.routes.js*” são as seguintes:

- Rota do registo do paciente: para o registo de um paciente é utilizado um validador para verificar os dados e, em seguida, é acedida a função “*create*” do controlador “*PatientController*”.
- Rota para atualizar paciente: é utilizado o mesmo validador da rota anterior para validar os dados e é executada a função “*update*” do controlador “*PatientController*”. Nesta rota é atualizada a informação do paciente editada pelo utilizador.
- Rota para obter todos os pacientes através do PS associado: através do *get* são obtidos todos os pacientes associados ao PS autenticado no *site*. Esta rota utiliza a função “*show*” do controlador “*PatientController*” para obter os pacientes.
- Rota para obter um paciente através do seu ID: no endereço da rota é enviado o ID do paciente para obter, através da função “*showById*” do controlador “*PatientController*”, a informação detalhada do paciente.

- Rota para obter todas as avaliações neuropsicológicas de um paciente: para todas as avaliações neuropsicológicas de um paciente serem solicitadas é necessário que o ID do paciente esteja no endereço da rota. A função “*showAllFromPatient*” do controlador “*PatientController*” é executada na rota.
- Rota do perfil cognitivo do paciente: esta rota *get*, contém no endereço o ID do paciente do qual se pretende visualizar o perfil cognitivo (performance absoluta) mais recente. Para tal, é utilizada a função “*showCognitiveProfileByPatientId*” do controlador “*CognitiveProfileController*”.
- Rota para calcular o perfil cognitivo do paciente: é um *post* no qual é calculado o perfil cognitivo do paciente consoante as avaliações neuropsicológicas e é inserido o mesmo na base de dados. A função do controlador “*CognitiveProfileController*” executada para calcular o perfil chama-se “*create*”.
- Rota da interpretação clínica: o endereço da rota contém o ID do paciente do qual se pretende visualizar a interpretação clínica mais recente, obtendo-se assim o perfil cognitivo através da função “*showInterpretacionClinicalByPatientId*” do controlador “*InterpretationClinicalController*”.
- Rota para calcular a interpretação clínica do paciente: é calculada a interpretação clínica consoante a sessão de avaliação neuropsicológica mais recente e os resultados da interpretação clínica nas avaliações utilizadas nessa sessão. Para tal, é acedida pela rota a função “*create*” do controlador “*InterpretationClinicalController*”.

Para obter-se as informações clínicas do paciente utilizou-se as rotas do ficheiro “*clinicalInfo.routes.js*”. As funções executadas na rota são do controlador “*ClinicalInformationController*”. As rotas neste ficheiro são as seguintes:

- Rota do registo da informação clínica do paciente: tal como as rotas de inserção de dados na base de dados apresentadas anteriormente é necessário um validador nesta rota para verificar os dados da informação clínica. A função utilizada é o “*create*” do controlador.
- Rota para atualizar as informações clínicas: quando o PS edita a informação clínica do paciente é utilizada esta rota para atualizar a informação. É utilizado o mesmo validador que a rota anterior para validação dos dados e a função a “*update*” do controlador.
- Rota para eliminar informação clínica: esta rota permite eliminar uma informação clínica selecionada pelo PS. É executada a função “*deleteClinicalInfo*” para eliminar essa informação.

O ficheiro “*patientAssessment.routes.js*” corresponde às rotas de todos os instrumentos de avaliação. Todas as funções executadas nas rotas seguintes estão implementadas no controlador “*PatientAssessmentsController*”. Para cada instrumento de avaliação existe uma rota de registo e uma de atualização. Deste modo só serão apresentadas as seguintes rotas de inserção e atualização dos instrumentos de avaliação MOCA e CDR:

- Rota para obter a avaliação neuropsicológica através do ID e do instrumento de avaliação: para obter uma avaliação neuropsicológica específica é necessário passar o ID da avaliação neuropsicológica e o nome da tabela do instrumento de avaliação através do endereço da rota. Esta rota devolve todas as informações da

avaliação neuropsicológica selecionada e os resultados do instrumento de avaliação utilizado na mesma, executando a função “*findOne*” do controlador.

- Rota de registo do instrumento de avaliação MOCA: são utilizados um validador para os dados inseridos na base de dados e a função “*createMOCA*” do controlador. Esta rota serve para registar os resultados do paciente no instrumento de avaliação MOCA.
- Rota para atualizar a avaliação realizada com o instrumento MOCA: tal como a anterior, é utilizado o mesmo validador para verificar os dados editados antes de serem atualizados na base de dados. É utilizada a função “*updateMOCA*” do controlador.
- Rota de registo do instrumento de avaliação CDR: o mesmo processo ocorre na rota de registo do instrumento CDR, em que existe um validador do instrumento para verificar os dados e é utilizada a função “*createCDR*” do controlador para inserção dos dados na base de dados.
- Rota para atualizar a avaliação realizada com o instrumento CDR: a rota utiliza o mesmo validador que a rota de registo do CDR e é utilizada a função “*updateCDR*” para atualizar os dados editados.
- Rota de eliminar avaliação: para eliminar a avaliação selecionada pelo PS utiliza-se esta rota, onde a mesma acede à função “*deleteAssessment*” para executar a eliminação dos resultados no instrumento de avaliação e também da avaliação.

As rotas relacionadas com os programas de treinos cognitivos foram implementadas no ficheiro “*trainingProgram.routes.js*”. As rotas apresentadas em seguida executam funções do controlador “*TrainingProgramController*”.

- Rota de registo do programa de treino: é utilizado um validador para verificar os dados antes da inserção de um programa de treino e a função “*create*” é executada.
- Rota para atualizar o programa de treino: tal como o registo, é utilizado o mesmo validador para verificar os dados editados do programa de treino e, em seguida, é executada a função “*update*”.
- Rota para obter todos os programas de treinos através do PS: esta rota devolve todos os programas de treinos associados ao PS autenticado, tendo em conta se o programa está concluído ou não. No endereço da rota é enviado o ID do PS e uma variável que define se os treinos que estão a ser solicitados são concluídos ou não, sendo esta “0” ou “1”. A função “*showAll*” é executada nesta rota.
- Rota para obter um programa de treino através do ID do mesmo: para solicitar informação sobre um programa de treino e todas as suas sessões de treino foi utilizado um *get*. O ID do programa de treino é enviado no endereço da rota e é executada a função “*findOne*” na mesma.
- Rota para obter todos os programas de treino de um paciente: no endereço da rota é enviado o ID do paciente, sendo este utilizado para obter todos os programas de treinos associados ao mesmo. Assim, é utilizada a função “*showAllByID*”.
- Rota para eliminar um programa de treino: caso o PS necessite de eliminar um programa de treino, a plataforma utiliza esta rota para eliminar o programa de treino e toda a informação associada ao mesmo. A função “*deleteTrainingProgram*” é executada nesta rota.

Ainda neste ficheiro (“*trainingProgram.routes.js*”), existem 9 rotas desenvolvidas que não são utilizadas no lado do *frontend*, mas sim por outros colegas no projeto, sendo estas:

- Rota de registo de uma sessão de treino cognitivo: utiliza-se um validador para verificar os dados antes da inserção da sessão de treino na base de dados e executa-se a função “*create*” do controlador “*SessionsTrainingController*”.
- Rota para atualizar uma sessão de treino cognitivo: apenas os campos da data final de sessão, do estado, da duração total, da performance e das notas são atualizados nesta rota, sendo que os restantes são inseridos nas rotas anteriores. É utilizada a função “*update*” do controlador “*SessionsTrainingController*”.
- Rota de registo das interações das atividades cognitivas: para esta rota é utilizada a função “*createIteration*” do controlador “*ActivitiesController*” para registar as interações das atividades cognitivas.
- Rota para atualizar a performance das interações das atividades cognitivas: é atualizada a performance, a data de conclusão da interação e o ID da sessão de treino de cada interação da atividade cognitiva. Esta atualização é realizada através do ID da interação, sendo que nesta rota é utilizada a função “*updateActivitiesInteractions*” do controlador “*ActivitiesController*”.
- Rota para remoção das interações das atividades cognitivas: utiliza-se a função “*deleteActivitiesInteractions*” do controlador “*ActivitiesController*” para eliminar da base de dados a interação de atividade pretendida através do seu ID.
- Rota para concluir o programa de treino: quando as sessões de treino se encontram concluídas, é necessário concluir o programa de treino através do seu ID. Deste modo utilizou-se a função “*updateIsDone*” do controlador “*TrainingProgram*”.
- Rota de registo dos eventos das atividades cognitivas: são registados os eventos das atividades cognitivas através da função “*createEvents*” do controlador “*ActivitiesController*”.
- Rota de registo de alarme: é utilizado um validador para verificar os dados do alarme a serem inseridos e é utilizada a função “*createAlarm*” do controlador “*AlarmsController*”.

Por fim tem-se o ficheiro “*formsFields.routes.js*” que contém todas as rotas que são utilizadas para obtenção dos dados dos campos dos formulários usados no *frontend*. Todas as funções do controlador executadas nas rotas são do controlador “*FormsController*”, sendo estas:

- Rota para os campos no registo do paciente: através desta rota são obtidos os dados utilizados para o formulário de registo do paciente, tais como: os níveis de escolaridade, os membros de família, o estado civil e a situação profissional. É utilizado a função “*getPatients*” do controlador.
- Rota para os campos no registo do utilizador: os únicos dados solicitados para o formulário de registo de utilizadores são as funções possíveis para um utilizador da plataforma (administrador, clínico ou investigador) e estes são obtidos através da função “*getRoles*” do controlador.
- Rota para os tipos de informação clínica: para obter todos os tipos de informação clínica para o formulário de registo das informações clínicas do paciente é utilizada a função “*getClinicalInfoTypes*” do controlador.

- Rota para os tipos de instrumentos de avaliação: para obter os dados para o campo de seleção do instrumento de avaliação, que serão inseridos os seus resultados, são solicitados os instrumentos de avaliação através da função “*getAssessmentTools*”.
- Rota para obter os dados dos campos no registo de uma avaliação neuropsicológica: posteriormente para a seleção do instrumento de avaliação, é necessário obter todos os campos de cada instrumento para sua inserção. Para tal utilizou-se a função “*getAssessmentFields*” do controlador. É enviado o ID do instrumento de avaliação pelo endereço da rota.
- Rota para as sessões de avaliação existentes: é enviado pelo endereço da rota o ID do paciente de forma a obter todas as sessões de avaliação existentes. Estes dados solicitados servem para o formulário de inserção de uma avaliação neuropsicológica, no campo das sessões existentes. É utilizada a função “*getAssessmentSessions*” do controlador.
- Rota para os tipos de interpretação clínica: esta rota é utilizada para obter todos os tipos de interpretação clínica (cognitiva e não cognitiva). Estes dados solicitados são utilizados no formulário de inserção de uma avaliação neuropsicológica. A função “*getAssessmentFieldsInterpretacionClinical*” do controlador é acedida pela rota.

4.3.4. Controladores

Os controladores manipulam os modelos de dados, e suas funções são acedidas nas rotas de forma a retornar dados conforme os pedidos. A figura abaixo (Fig. 4.8) apresenta a organização dos controlares na API.

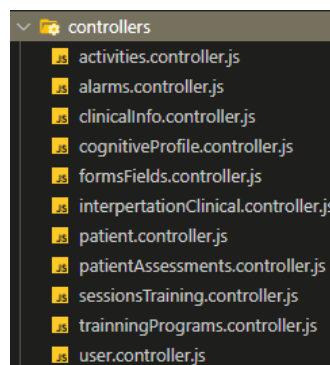


Fig. 4.8 – Organização dos controladores na API.

Existem algumas configurações em comum em todos os controladores, tais como: a importação do *dotenv*, a execução da função *config()* no *dotenv*, a importação dos modelos necessários para cada controlador e a importação da classe *HttpExceptionI*. Esta última classe estende a classe *Error* e contém um construtor com os parâmetros *status*, *message* e *data*. A função *checkValidation* está presente em diversos controladores e contém a seguinte estrutura apresentada na Fig. 4.9.

```

checkValidation = (req) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    throw new HttpException(400, error.errors[0].msg, error);
  }
}

```

Fig. 4.9 – Função “*checkValidation*”.

Quando é realizada a validação dos dados antes da inserção ou atualização na base de dados, nas funções assíncronas dos controladores, a função “*checkValidation*” verifica se ocorreram erros na validação dos dados e, se for o caso, lança uma exceção com o estado, a mensagem do erro e o erro.

Os diversos controladores apresentados na Fig. 4.8 podem ser sumarizados como:

- Controlador das atividades cognitivas (ficheiro “*activities.controller.js*”): existem três funções dentro do controlador, sendo estas: a criação de uma iteração de atividade, a criação de um evento, a atualização das interações das atividades e a remoção de uma iteração de atividade cognitiva.
- Controlador dos alarmes (ficheiro “*alarms.controller.js*”): para enviar uma notificação para o *frontend* existe a função de criação de um alarme neste controlador, sendo esta utilizada por outros colegas.
- Controlador da informação clínica (ficheiro “*clinicalInfo.controller.js*”): as funções existentes no controlador são a criação, a atualização e a eliminação da informação clínica do paciente.
- Controlador do perfil cognitivo (ficheiro “*cognitiveProfile.controller.js*”): as funções principais deste controlador são: encontrar o perfil cognitivo mais recente do paciente solicitado e a criação do perfil cognitivo. Além disso no controlador existem mais 10 funções auxiliares que ajudam na criação do perfil cognitivo, sendo que a principal é a do cálculo do perfil cognitivo através de um algoritmo desenvolvido por um elemento do projeto. Esta função verifica quais os instrumentos de avaliação que existem na sessão de avaliação mais recente e percorre cada um calculando os domínios (cognição geral, memória, funções executivas, linguagem e atenção) consoante o modelo desenvolvido pelas PS do projeto. As funções secundárias são a obtenção do valor relativo, e funções que calculam os resultados de cada instrumento apresentado em seguida: MOCA, CDR, FCSRT, WAIS-II, Fluências Verbais Semânticas e Fonéticas, Figura Complexa de Rey, Toulouse-Piéron e WMS.
- Controlador dos campos dos formulários (ficheiro “*formsFields.controller.js*”): é possível obter informações para certos campos dos formulários (registo do paciente, do utilizador, da informação clínica e das avaliações neuropsicológicas) existentes no *frontend* através das funções do controlador.
- Controlador da interpretação clínica (ficheiro “*interpertationClinical.controller.js*”): as funções principais deste controlador são: a obtenção da interpretação clínica mais recente do paciente e a criação da interpretação clínica com dados da interpretação inseridos na sessão de avaliação neuropsicológica mais recente, onde é realizado o mesmo processo de cálculo do perfil cognitivo.
- Controlador dos pacientes (ficheiro “*patient.controller.js*”): as funções existentes são: a criação e a atualização da informação pessoal e sociodemográfica do paciente, a obtenção da informação através da identificação do paciente, a

obtenção de todos os pacientes associados ao utilizador autenticado, a obtenção das avaliações neuropsicológicas dos pacientes, a obtenção de todos os pacientes não associados a nenhum PS e a criação e a eliminação da associação do paciente e do PS.

- Controlador das avaliações neuropsicológicas do paciente (ficheiro “*patientAssessments.controller.js*”): contém a função para encontrar uma avaliação neuropsicológica através da sua identificação e do nome do instrumento de avaliação, sendo que, para todos os instrumentos de avaliação, existe uma função de criação e de atualização. A última função do controlador é a eliminação de uma avaliação neuropsicológica e todos os seus dados associados.
- Controlador das sessões de treino (ficheiro “*sessionsTraining.controller.js*”): as funções de criação e de atualização de uma sessão de treino estão presentes neste controlador.
- Controlador dos programas de treino (ficheiro “*trainingPrograms.controller.js*”): contém as funções de criação, atualização e eliminação dos programas de treino, de obtenção de: todos os programas ativos/concluídos associados ao utilizador autenticado e se estão ativos/concluídos, de toda a informação de um programa de treino através da sua identificação, de todos os programas de treino associados a um paciente, e de atualização de um campo do programa de treino quando o mesmo é concluído.
- Controlador dos utilizadores (ficheiro “*user.controller.js*”): é possível obter os utilizadores, a informação específica de um utilizador, a criação e a atualização de um atualizador e o *login* através das funções definidas neste controlador.

A estrutura das funções nos controladores é semelhante à estrutura da função apresentada na Fig. 4.6.

4.3.5. Modelos

Os modelos desenvolvidos no projeto servem para aceder e/ou enviar dados para a base de dados. Na Fig. 4.10 apresenta-se a organização dos modelos na API.

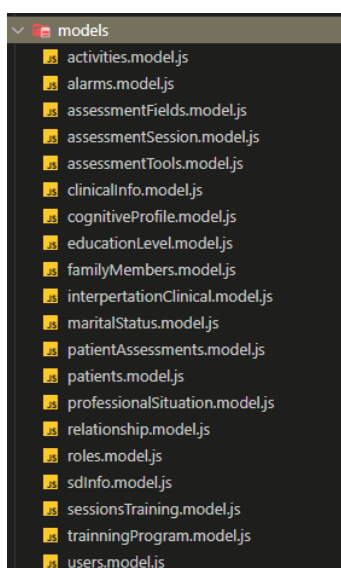


Fig. 4.10 – Organização dos modelos na API.

Do mesmo modo que os controladores, nos modelos também existem algumas configurações em comum, tais como: a importação do módulo da conexão da base de dados (apresentado na secção 4.3.1.1), a importação da função *multipleColumnSet* e da função *checkDate* quando necessárias, a importação do módulo *constants* (apresentado na secção 4.3.1.3) e algumas funções de alguns modelos quando as mesmas eram utilizadas.

A função *multipleColumnSet* recebe como parâmetro um objeto JSON com chave e o valor, sendo a chave o nome do campo na base de dados e o dado correspondente ao campo sendo o valor. As chaves e os valores são transformados em dois novos objetos JSON correspondentes. É criado um *array columnSet* que percorre o objeto Json das chaves e adiciona a cada chave os caracteres “=?”, separando a seguinte chave com o caractere “;”. É retornado o *columnSet* e o objecto JSON dos valores. Esta função é utilizada nas funções de *update* nos modelos, devido a estrutura da *query sql* de um *update*. A função *checkDate* recebe como argumento uma data e retorna se a mesma é válida.

Os modelos comunicam diretamente com a base de dados e, assim sendo, os modelos apresentados na Fig. 4.10 contêm todas as *querys* desenvolvidas para a obtenção, o envio, a atualização e a eliminação de dados. A estrutura das funções nos modelos é semelhante à estrutura da função apresentada na Fig. 4.6. Na Fig. 4.11 a) e b) estão apresentados dois exemplos de funções de modelos presentes na API.

```
//FindOne patient with Patient ID
findOne = async (params) => {
  const { columnSet, values } = multipleColumnSet(params)
  const sql = `SELECT * FROM \`${constants.tablePatients}\` WHERE ${columnSet}`;
  const result = await db.query(sql, [...values]);
  const { password, ...userWithoutPassword } = result[0];
  return userWithoutPassword;
}
```

a)

```
//create patient
create = async ({ name, email, contactNumber, address, job, date_of_birth, sex, maritalStatus, education_level_id, family_members, creationData,
  associatedHealthProfessional, isActive = 1, professionalSituation, laterality }) => {
  try {
    const conn = await db.trans.getConnection();
    const success = await transaction(db.trans, async conn => {
      let sql = `INSERT INTO \`${constants.tablePatients}\` (\`email\`,\`isActive\`,\`contactNumber\`,\`name\`,\`address\`,\`password\`) VALUES (?, ?, ?, ?, ?)`;
      let result = await conn.query(sql, [email, isActive, contactNumber, name, address, "temp"]);
      const patId = result[0].insertId;

      sql = `INSERT INTO \`${constants.tableSociodemographicData}\` (\`creationData\`,\`Patients_id\`,\`Education_levels_id\`,\`date_of_birth\`,\`sex\`,\`job\`,
        \`MaritalStatus_id\`,\`ProfessionalSituation_id\`,\`laterality\`) VALUES (?, ?, ?, ?, ?, ?, ?, ?)`;
      result = await conn.query(sql, [creationData, patId, education_level_id, date_of_birth, sex, job, maritalStatus, professionalSituation, laterality]);

      family_members.forEach(element => {
        sql = `INSERT INTO \`${constants.tableFamilyMember}\` (\`Relationship_id\`,\`Patients_id\`,\`creationData\`) VALUES (?, ?, ?)`;
        conn.query(sql, [element, patId, creationData]);
      });

      sql = `INSERT INTO \`${constants.tablePatientsHealthProfessional}\` (\`Patients_id\`,\`HealthProfessional_id\`) VALUES (?, ?)`;
      result = await conn.query(sql, [patId, associatedHealthProfessional]);
    });
    return success;
  } catch (error) {
    return error;
  }
}
```

b)

Fig. 4.11 – Modelo "PatientModel": a) função "findOne" – query simples; b) função "create" - transação.

Observando a Fig. 4.11 a) tem-se a função “*findOne*” que contém uma *query* simples à tabela dos pacientes. Esta função retorna todas informações do paciente solicitado, menos a *password*. A Fig. 4.11 b) apresenta a função “*create*” que realiza a inserção de todas as informações do novo paciente nas tabelas correspondentes, sendo esta função uma transação complexa. Os dados são inseridos na tabela do paciente, na da informação sociodemográfica, na dos membros de família e na da associação do paciente e profissional de saúde. Em caso de falha nenhuma das informações ficam guardadas na base de dados, evitando-se a redundância de dados.

4.3.6. *Middlewares* e validadores

Os *middlewares*¹² no Express auxiliam a executar qualquer código, a realizar modificações nas solicitações e respostas, a terminar com as tarefas de solicitação e pedido, a executar a próxima função na pilha, entre outros. Os validadores implementados são feitos através da importação do módulo *express-validation*¹³, que em caso de falha na validação definida, retorna o erro obtido.

Na Fig. 4.12 estão apresentados os *middlewares* e os validadores organizados na API.

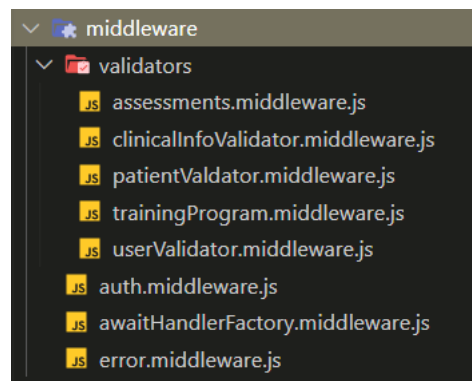


Fig. 4.12 – Organização dos *middlewares* e validadores na API.

Em relação ao *middleware awaitHandleFactory*, este já não será referido aqui, visto que já foi apresentado na secção 4.3.3. O *middleware error* contém a função “*errorMiddleware*” que, quando ocorra algum erro no servidor, é enviada uma mensagem que ocorreu um erro interno e o estado de erro. Esta função é importada no ficheiro *server.js* apresentado na secção 4.3.1.2.

No *frontend* é feita a validação dos campos através das propriedades dos *inputs* (por exemplo, obrigatoriedade de ser preenchido, mínimo e máximo, ser email, entre outros), mas para certificar que não existem dados inválidos a serem inseridos na base de dados foram desenvolvidos validadores, sendo estes:

- Validador das avaliações neuropsicológicas (ficheiro “*assessments.middleware.js*”): existe uma função que contém os dados em comum na inserção de todos os instrumentos de avaliação. Assim sendo é realizada a validação no ID do paciente e no ID do PS. Para cada instrumento

¹² **Middleware no Express** - <http://expressjs.com/en/guide/using-middleware.html>

¹³ **express-validation**, <https://www.npmjs.com/package/express-validation>

existe uma função de validação que verifica se o campo existe, se o valor do campo está entre o mínimo e o máximo, entre outros.

- Validador da informação clínica do paciente (ficheiro “*clinicalInfoValidator.middleware.js*”): validação dos dados na inserção e na atualização de uma informação clínica associada a um paciente.
- Validador da informação do paciente (ficheiro “*patientValidator.middleware.js*”): contém apenas uma função de validação quando é registado um novo paciente na base de dados que valida, por exemplo o *email*, o número de telemóvel, a data de nascimento, entre outros.
- Validador do programa de treino (ficheiro “*trainingProgram.middleware.js*”): tal como nos validadores anteriores, neste existe uma função que valida os dados inseridos no formulário do programa de treino. O mesmo validador verifica os dados inseridos para criação e para atualização de uma sessão de treino.
- Validador da informação do utilizador (ficheiro “*userValidator.middleware.js*”): existem 2 funções que distinguem o validador da criação de um utilizador e da atualização da sua informação. O da criação valida todos os dados inseridos no formulário de criação de um utilizador, pelo administrador. O de atualização valida os dados permitidos o próprio atualizar no seu perfil, sendo estes: o nome, o email, a cédula profissional e o número de contato.
- Validador da interação das atividades cognitivas: para a inserção de uma iteração das atividades cognitivas são necessárias as validações dos dados recebidos correspondente à iteração e também os dados recebidos para a inserção da tabela de definição de variáveis em cada interação.

Em caso de erro na validação num dado é enviada uma mensagem com o nome do campo que não cumpre com a validação. Devido ao facto de os dados registados no NexusBRaNT poderem ser editados, os validadores de algumas funções de criação são os mesmos utilizados nas funções de atualização.

4.4. *Frontend*

O *frontend* implementado no projeto teve como principal objetivo o desenvolvimento de uma plataforma *web* em React (referida anteriormente na secção 2.3.4) para a gestão de pacientes e personalização de treinos cognitivos. Tal como no *backend*, este segue o padrão arquitetural MVC, sendo que a “*view*” está implementada no *frontend*. Os PS têm acesso direto à plataforma, sendo esta dividida posteriormente em duas: o acesso como administrador e como utilizador comum (PS clínico ou PS investigador).

Na Fig. 4.13 está apresentada a organização do NexusBRaNT.

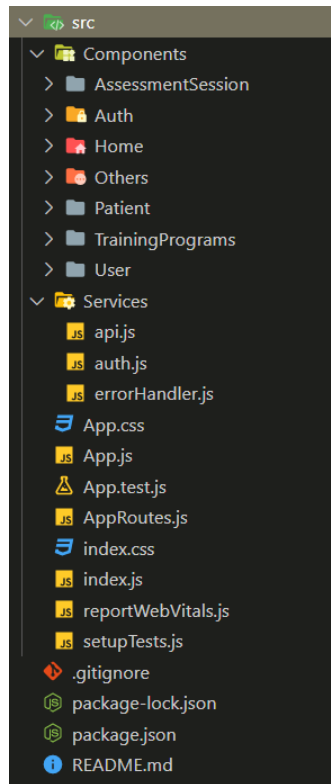


Fig. 4.13 – Organização do NexusBRaNT.

Observando a Fig. 4.13 é possível verificar três grupos principais: componentes, serviços e configurações da *app*. Dentro do grupo dos componentes existem sete subgrupos que contêm as *views* do *website*, sendo estes: avaliações neuropsicológicas, autenticação, *home*, componentes reutilizáveis, paciente, programas de treino cognitivo e utilizadores. Na pasta dos serviços contém ficheiros com configurações essenciais para o acesso a API, autenticação dos utilizadores e notificação de erros. Por último, os ficheiros de configurações contêm as inicializações da aplicação e as rotas internas da mesma. Nas secções seguintes será apresentado com descrição os grupos descritos anteriormente.

4.4.1. Visão geral

Uma arquitetura de interface de utilizador pode ser apresentada através de um *dialog map*, sendo que este apresenta os elementos de diálogo que o utilizador interage e as navegações permitidas entre os elementos, sendo uma visão de alto nível de abstração da interface de utilizador [51]. Considerando a complexidade e a quantidade de funcionalidades implementadas ao longo de todo o projeto tem-se apresentado na Fig. 4.14 um *dialog map* do funcionamento geral da plataforma *web*.

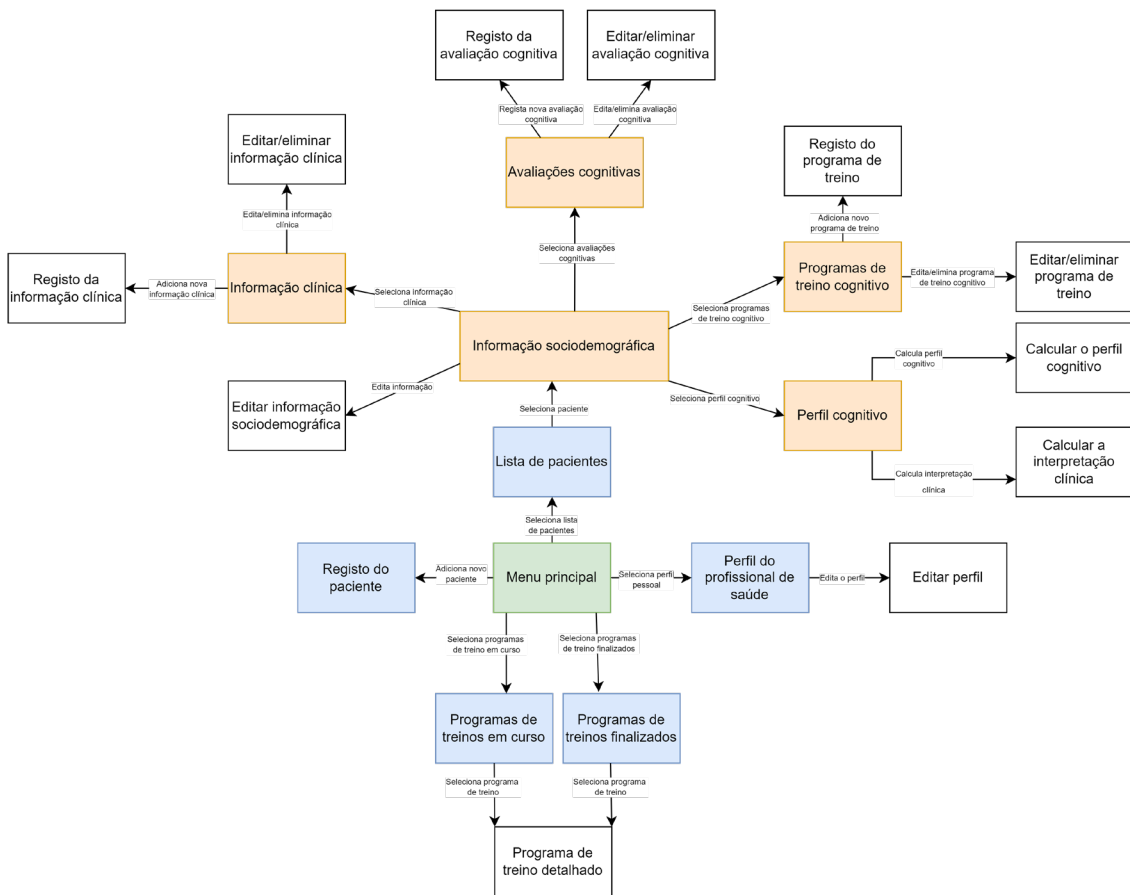


Fig. 4.14 – *Dialog map* geral do NexusBRaNT.

Através da Fig. 4.14 é possível observar as diversas funcionalidades que os PS poderão executar no *site*. Tem-se a verde o menu principal que contém as seguintes funcionalidades:

- Listagem dos pacientes: ao selecionar um paciente da lista obtém-se uma página com a informação sociodemográfica. A partir da mesma é possível navegar para as outras funcionalidades (em amarelo na Fig. 4.14), sendo estas: informação clínica, avaliações neuropsicológicas, programas de treino cognitivo e perfil cognitivo.
- Perfil do PS: é possível obter todas as informações do utilizador autenticado na plataforma.
- Registo do paciente: é nesta página que o utilizador regista um novo paciente.
- Programas de treinos em curso e programas de treinos finalizados: apesar das duas funcionalidades serem distintas, é utilizada a mesma página para apresentar a informação detalhada do programa de treino selecionado, tendo algumas variações entre eles.

Na Fig. 4.15 é apresentado o *dialog map* das funcionalidades que apenas o administrador do NexusBRaNT poderá aceder.

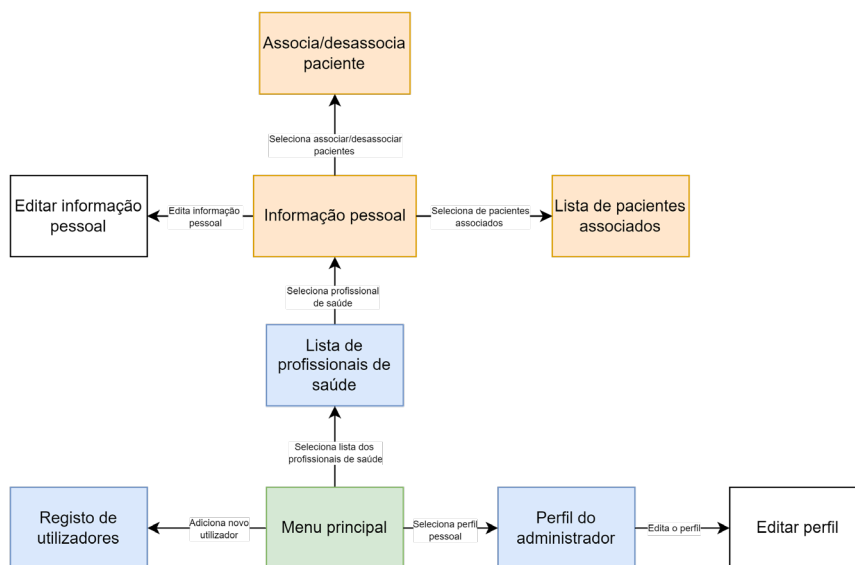


Fig. 4.15 – *Dialog map* da gestão do NexusBRaNT.

Tal como no *dialog map* anterior (Fig. 4.14), é possível observar na Fig. 4.15 a existência do menu principal que contém como funcionalidades essenciais (em azul):

- Registo de utilizadores: é possível através desta funcionalidade registar um novo utilizador.
- Perfil do administrador: é possível visualizar os dados pessoais do administrador autenticado e editá-los.
- Lista de profissionais de saúde: ao selecionar o PS da lista é possível obter a sua informação pessoal, sendo que a partir da mesma é possível navegar para as outras funcionalidades em amarelo, sendo estas: lista de pacientes associados e associar/desassociar de paciente.

As funcionalidades apresentadas em branco nas Fig. 4.14 e Fig. 4.15 são subfuncionalidades das apresentadas em azul e laranja. A partir de cada página é possível voltar à anterior como também ao menu principal, pois este se encontra na lateral esquerda de todo o *site*, independentemente se o acesso é realizado por um utilizador ou um administrador. Posteriormente serão apresentadas e descritas todas as páginas das funcionalidades contidas na Fig. 4.14 e Fig. 4.15.

4.4.2. Comunicação frontend/backend

Utilizou-se a biblioteca *axios*¹⁴ para realizar a ligação entre o *frontend* e o *backend*. Esta biblioteca utiliza AJAX para pedidos assíncronos. Para tal, no ficheiro “*api.js*” é realizado a importação da biblioteca, a definição numa constante do *url* onde contém a API que está a ser executada e a criação da instância do *axios.create()* para uma configuração customizada sempre que é necessário utilizar uma rota.

¹⁴ **axios**, <https://www.npmjs.com/package/axios>

Ainda dentro do ficheiro “*api.js*” encontra-se implementadas todas as funções de envio, pedido, remoção e atualização dos dados conforme o desenvolvimento das *views*. São estas funções que comunicam com a API indicando quais as rotas da mesma que serão executadas.

Tendo em conta que as estruturas das funções apresentadas anteriormente são muito semelhantes entre si, serão apresentadas na Fig. 4.16 a) e b) dois exemplos de um *post* e de um *get*, respetivamente.

```
export const registerPatient = async (name, email, contactNumber, address, job, date_of_birth, sex, maritalStatus, education_level_id, family_members, creationData, associatedHealthProfessional, professionalSituation, laterality) => {  
  
  let errorFlag = false;  
  let data = {};  
  await axiosInstance.post(baseUrl + "/patients", { name, email, contactNumber, address, job, date_of_birth, sex, maritalStatus, education_level_id, family_members, creationData, associatedHealthProfessional, professionalSituation, laterality })  
  .catch(function (error) {  
    if (error.response) {  
      errorFlag = true;  
      data = error.response;  
    } else if (error.request) {  
      errorFlag = true;  
      data = error.response;  
    } else {  
      errorFlag = true;  
      data = error.response;  
    }  
  });  
  if (!errorFlag) { console.log("Paciente registado na base de dados"); }  
  return { error: errorFlag, data: data };  
};
```

a)

```
export const getPatientbyID = async (idPatient) => {  
  return axiosInstance.get(baseUrl + '/patients/' + idPatient);  
};
```

b)

Fig. 4.16 – Funções no ficheiro “*api.js*”: a) *registerPatient* – *POST*; b) *getPatientbyID* – *GET*.

Na Fig. 4.16 a) está apresentada a função “*registerPatient*”, sendo esta responsável por enviar os dados para a inserção de um novo paciente na base de dados. É executada uma solicitação do tipo *post* onde é indicado o *url* (referido anteriormente) que contém a API, a rota correspondente e os dados registados pelo PS. É utilizada a função de *catch* para obter os erros caso estes ocorram. É retornado um objeto *json* com a informação se ocorreu erro e os dados do mesmo.

A função “*getPatientsbyID*” apresentada na Fig. 4.16 b) é um exemplo simples de uma solicitação do tipo *get* que tem como parâmetro o ID do paciente e retorna à função *axios* onde é definido novamente o *url* para a API e a rota correspondente da mesma.

4.4.3. Design

A linguagem visual do NexusBRaNT teve como base o *design* anteriormente desenvolvido, sendo que o mesmo foi fornecido por colegas. Tendo em conta todos os pontos fortes e fracos (referidos na secção 2.2.3), foi necessário realizar alterações no *design* para a nova plataforma. Deste modo selecionou-se as duas cores predominantes no *site* tendo em conta o logotipo criado pelo projeto BRaNT. Começou-se por criar a interface dos formulários e *login* e depois solicitou-se *feedback* por parte dos PS de forma a prosseguir com o *design* escolhido.

No início do processo de elicitação de requisitos foi indicado pelos PS que o *design* fosse intuitivo, de fácil utilização e minimalista. Considerando os requisitos apresentados e a escolha do React para a implementação da plataforma, optou-se por desenvolver um layout de página única (*Single Page Application*). Como referido anteriormente (na secção 4.1), a biblioteca utilizada para o *design* foi a Material UI.

Quando não foi possível aceder às funcionalidades do *website* anterior, para ter uma base para o *design*, recorreu-se as PS de forma a descreverem a apresentação e organização pretendidas. Desta forma criou-se um protótipo que foi apresentado às mesmas antes da implementação final da funcionalidade a ser desenvolvida. Por exemplo, nos menus dos programas de treino e suas secções foi necessário recorrer a um protótipo em papel para definir o *design* pretendido, recolha de *feedback* e definição final.

Durante o processo de desenvolvimento do NexusBRaNT as PS e outros elementos do projeto deram *feedback* sobre o *design* e, consoante o mesmo, eram efetuadas as alterações necessárias.

4.4.4. Componentes customizáveis

Para o desenvolvimento deste projeto foi essencial a utilização de componentes customizáveis, sendo estas uma das maiores vantagens da implementação em React. Para tal, utilizou-se componentes da biblioteca Material UI (tal como referido na secção 4.1) e através dos mesmos modificou-se conforme as necessidades das funcionalidades.

Os componentes simples podem se dividir em quatro grupos: campos dos formulários, visualização de informação, visualização de notificações e barra de navegação.

Na Fig. 4.17 estão apresentados todos os componentes existentes.

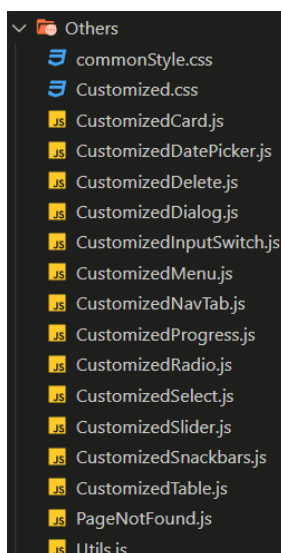


Fig. 4.17 – Componentes customizáveis no NexusBRaNT.

É possível observar na Fig. 4.17 os ficheiros “*CustomizedDatePicker*”, “*CustomizedInputSwitch*”, “*CustomizedRadio*”, “*CustomizedSelect*” e “*CustomizedSlider*”, sendo que estes são utilizados nos campos dos formulários. O “*CustomizedDatePicker*” é o componente utilizado no campo da seleção da data, sendo

que o mesmo está em diversos formulários ao longo do *site*, como por exemplo no registo do paciente, no registo da avaliação neuropsicológica, no registo de um programa de treino, entre outros. O componente “*CustomizedInputSwitch*” contém um *switch* que conforme o tipo de *input* apresenta um campo de formulário com todas as características correspondentes para que o PS insira o solicitado. Um dos exemplos é nas avaliações neuropsicológicas, onde este componente é usado para apresentar os diversos campos dos instrumentos de avaliação. Deste modo é apenas utilizado um único componente que é dinâmico e varia consoante o tipo de instrumento de avaliação selecionado.

Na Fig. 4.18 estão apresentados dois exemplos de inserção de avaliações neuropsicológicas com distintos instrumentos. É possível observar na figura o componente referido anteriormente (*CustomizedInputSwitch*), com mesmas características, contudo adaptado aos campos de cada instrumento, através da informação recebida sobre do campo. Na Fig. 4.18 a) tem-se a inserção de uma avaliação neuropsicológica do instrumento Rey15 e na Fig. 4.18 b) a inserção com o instrumento de avaliação IAFAI .

The figure consists of two side-by-side screenshots of a web form titled "Nova avaliação neuropsicológica : Érica Filipa Dias Cunha".

Form (a) - Rey15:

- Header: "Nova avaliação neuropsicológica : Érica Filipa Dias Cunha"
- Form fields:
 - Sessão de avaliação neuropsicológica :
 - Nova sessão
 - Sessão existente
 - Selecione o instrumento de avaliação: Rey15
 - 1 - Ensaio da evocação imediata (Rey15): Min:0 Max:15 *
 - 2 - Resultado combinado do reconhecimento: Min:0 Max:30 *
 - Observações: (text area)
 - Submeter avaliação (button)

Form (b) - IAFAI:

- Header: "Nova avaliação neuropsicológica : Érica Filipa Dias Cunha"
- Form fields:
 - Sessão de avaliação neuropsicológica :
 - Nova sessão
 - Sessão existente
 - Selecione o instrumento de avaliação: IAFAI
 - IAFAI 1 - ABVD: Min:0 Max:100 *
 - 2 - AIVD-F: Min:0 Max:100 *
 - 3 - AIVD-A: Min:0 Max:100 *
 - 4 - Total: Total * 0
 - Observações: (text area)
 - Submeter avaliação (button)

Fig. 4.18 – Nova avaliação neuropsicológica: a) Rey15; b) IAFAI.

O “*CustomizedRadio*” é utilizado para selecionar uma das opções apresentadas, sendo do tipo *input radio*, e tem-se como exemplo a seleção da sessão de avaliação, em que esta poderá ser nova ou existente, no registo de avaliações neuropsicológicas. O componente “*CustomizedSelect*” recebe da base de dados os valores que deverão ser apresentados para o PS selecionar, e conforme o tipo de dados, o valor selecionado é armazenado. Este componente é um dos mais utilizados devido ao facto de existirem diversos campos de seleção ao longo dos formulários, como por exemplo: situação profissional, nível de escolaridade, estado civil, sessões de avaliação, entre outros. O “*CustomizedSlider*” é o componente que contém um *slider* que permite selecionar o intervalo de valores no

formulário, neste caso o mesmo é utilizado no registo de um novo programa de treino cognitivo.

Na visualização de informação existem os componentes “*CustomizedCard*”, “*CustomizedDialog*”, “*CustomizedProgress*” e “*CustomizedTable*”, apresentados na Fig. 4.17. O componente “*CustomizedCard*” é utilizado para apresentar os dados dos programas/sessões de treino cognitivo em caixas (*boxes*) de texto definidas. No mesmo componente é chamado o componente “*CustomizedProgress*” onde definiu-se o formato do progresso dos programas de treino cognitivo. O “*CustomizedDialog*” é um componente que abre um *pop-up*, apresentando a informação detalhada de várias subcategorias. Tem-se como exemplo das subcategorias a informação clínica, avaliações neuropsicológicas, programas de treino cognitivo, entre outros, em que nestas é necessário seleccionar a informação detalhada a ser visualizada.

Na Fig. 4.19 estão apresentados dois exemplos do componente anterior (*CustomizedDialog*) em contextos distintos, sendo que na primeira (Fig. 4.19 a)) a informação clínica detalhada e na segunda (Fig. 4.19 b)) a informação do programa de treino cognitiva detalhada.

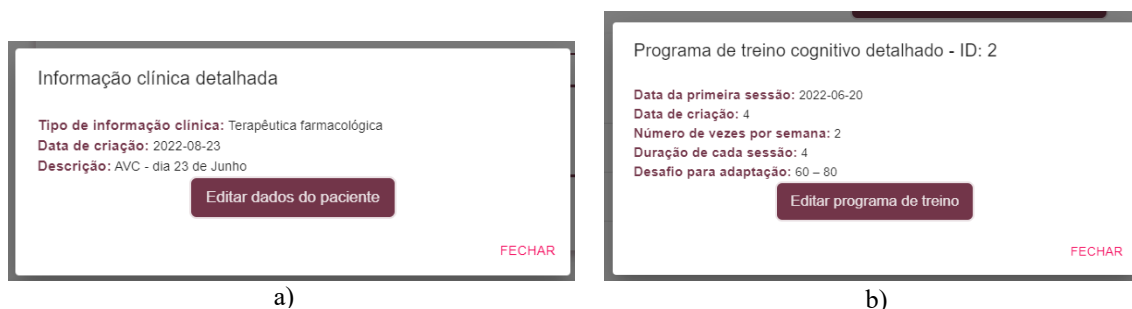


Fig. 4.19 – Informação detalhada: a) clínica; b) programa de treino cognitivo.

O “*CustomizedTable*” contém como componente uma tabela no qual, conforme a informação recebida, a mesma é modificada. Num dos exemplos da tabela é possível ocultar a barra de pesquisa e os filtros de ordenação, sendo que este ocorre na visualização de dados sociodemográficos e perfil cognitivo do paciente. Outro dos casos é quando é possível a pesquisa e os filtros de ordenação, define-se as colunas que serão apresentadas na mesma, sendo que os dados apresentados provêm da base de dados.

A Fig. 4.17 apresenta também o componente “*CustomizedSnackBar*” que tem como principal funcionalidade a visualização de notificações consoante a sua gravidade. Este componente é utilizado em quase todas as funcionalidades de forma a notificar se ocorreu um erro, um aviso ou um sucesso na tarefa realizada.

O componente “*CustomizedNavTab*” é utilizado como barra de navegação, este contém as divisões e o painel definidos quando são utilizados. Tem-se como exemplo as informações do paciente, no qual é possível aceder, através da barra de navegação, aos dados do paciente, informação clínica, avaliações neuropsicológicas, programas de treino e perfil cognitivo (Fig. 4.20 a)) e também às informações do PS (Fig. 4.20 b)), onde contém apenas acesso aos dados do PS.

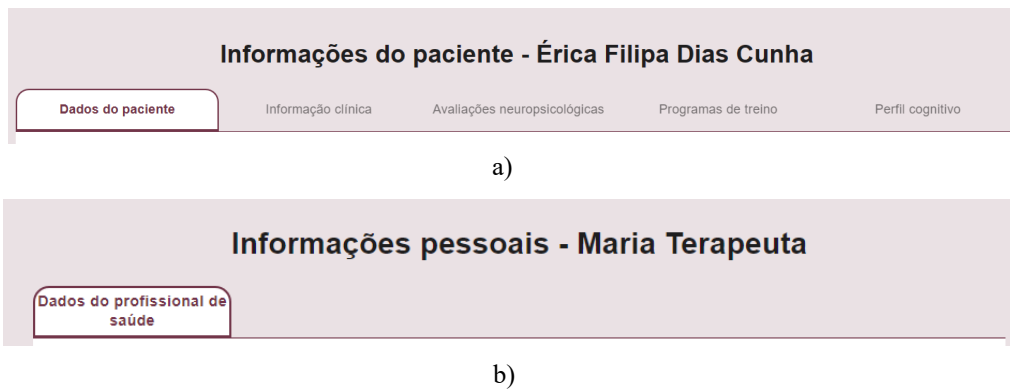


Fig. 4.20 – Barra de navegação: a) paciente; b) PS.

Na Fig. 4.17 está apresentado o componente “*CustomizedDelete*” que é utilizado para eliminação da informação selecionada. Neste componente é questionado ao utilizador se tem certeza que pretende eliminar a informação selecionada, sendo que, caso a resposta seja positiva, é eliminada a informação, caso seja negativa, o componente é encerrado e volta para a visualização da informação selecionada detalhada.

Através destes componentes e outros mais simplificados é possível construir componentes mais complexos, tais como os menus e os formulários, que estão presentes ao longo da plataforma. Nos menus tem-se somente um único componente desenvolvido que através dos parâmetros quando o mesmo é executado, este modifica-se conforme as necessidades impostas. Por exemplo, no menu do utilizador comum, não é possível visualizar todas as funcionalidades apresentadas quando um administrador está autenticado e vice-versa. Na Fig. 4.21 são apresentados os dois tipos de menus existentes na plataforma desenvolvida, sendo que consoante o utilizador autenticado o mesmo é alterado.



Fig. 4.21 – Menu: a) utilizadores em comum; b) administrador.

Nos formulários implementados foram desenvolvidos, inicialmente, para o registo dos dados que o PS. Mas durante o desenrolar do projeto e conforme as necessidades das PS, foi necessário a possibilidade de edição dos dados inseridos. Para tal, reformulou-se

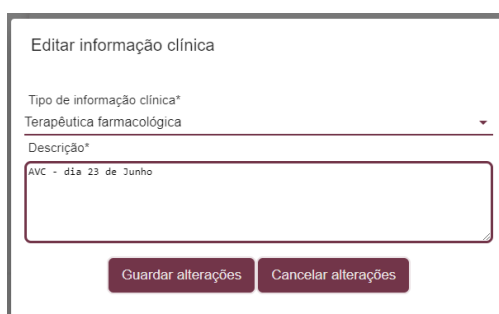
os componentes dos formulários criados de forma a utilizar os mesmos para a edição dos dados.

Na Fig. 4.22 está apresentado um exemplo do formulário de criação/edição da informação clínica.



Formulário de criação de informação clínica. O formulário contém um campo de seleção rotulado "Tipo de informação clínica*" com uma seta para baixo. Abaixo dele, há um campo de texto rotulado "Descrição*" que está atualmente vazio. Na base do formulário, há um botão azul com o texto "Registrar informação clínica".

a)



Formulário de edição de informação clínica. O formulário contém um campo de seleção rotulado "Tipo de informação clínica*" com o valor "Terapêutica farmacológica" selecionado. Abaixo dele, há um campo de texto rotulado "Descrição*" que contém o texto "AVC - dia 23 de Junho". Na base do formulário, há dois botões azuis: "Guardar alterações" e "Cancelar alterações".

b)

Fig. 4.22 – Formulário: a) criação; b) edição.

Na Fig. 4.22 a) é possível observar o formulário de criação de uma informação clínica, sendo que na Fig. 4.22 b) é visível o mesmo formulário já com a informação apresentada nos seus campos. Evita-se a redundância de código nos exemplos apresentados acima (componente de menu, Fig. 4.21, e formulário de informação clínica, Fig. 4.22). Deste modo tem-se a personalização de cada componente consoante as suas necessidades.

4.4.5. Funcionalidades

Neste subcapítulo é apresentada toda a interface com o utilizador. Este será dividido em dois, tendo em conta as funcionalidades apresentadas em cada tipo de utilizador (clínicos/investigadores e administrador). Ao longo da apresentação da interface serão referidas funcionalidades mais concretas da plataforma que ainda não foram apresentadas e, desta forma, também as interações entre as funcionalidades implementadas.

A funcionalidade de iniciar sessão na plataforma desenvolvida é em comum aos dois tipos de utilizadores. Consoante o utilizador autenticado a plataforma encaminha para as páginas correspondentes. Caso um utilizador tente se autenticar e não esteja na base de dados é apresentada uma mensagem de erro que indica que as credenciais de *login* estão incorretas.

Na Fig. 4.23 está apresentada a página de início de sessão.

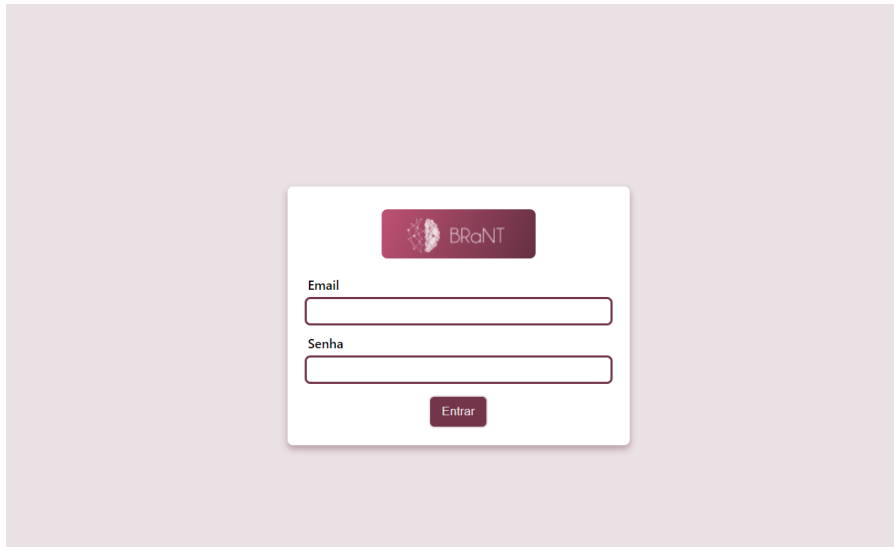


Fig. 4.23 – Funcionalidade de autenticação.

4.4.5.1. Utilizadores comuns (PS clínico/investigador)

Os utilizadores comuns têm acesso à gestão dos pacientes, avaliações neuropsicológicas e programas de treinos cognitivos, logo, após a autenticação, o utilizador é encaminhado para o ecrã de listagem dos seus pacientes, sendo este apresentado posteriormente. Tal como apresentado no menu (Fig. 4.21 b)) o PS consegue registar um novo paciente, estando esta funcionalidade apresentada na Fig. 4.24.

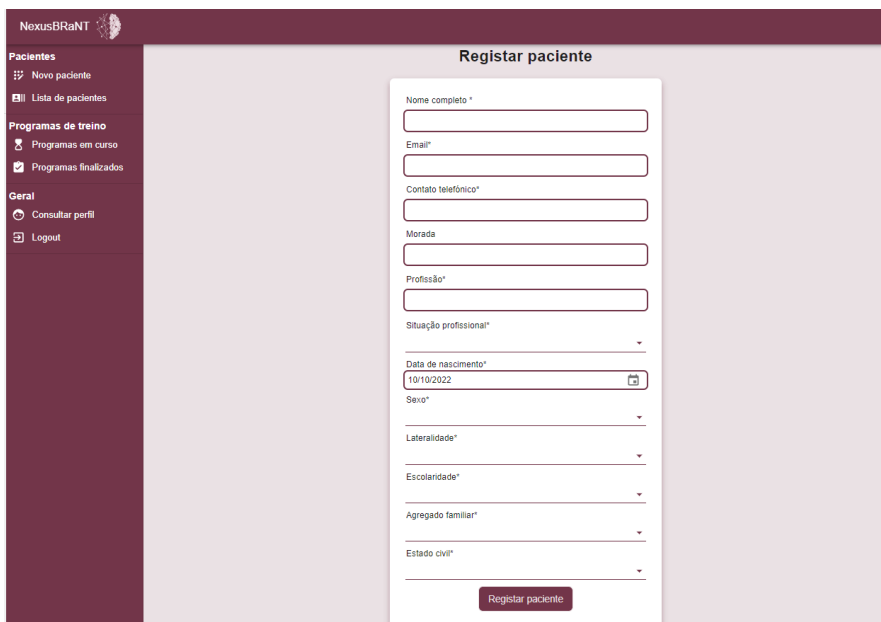
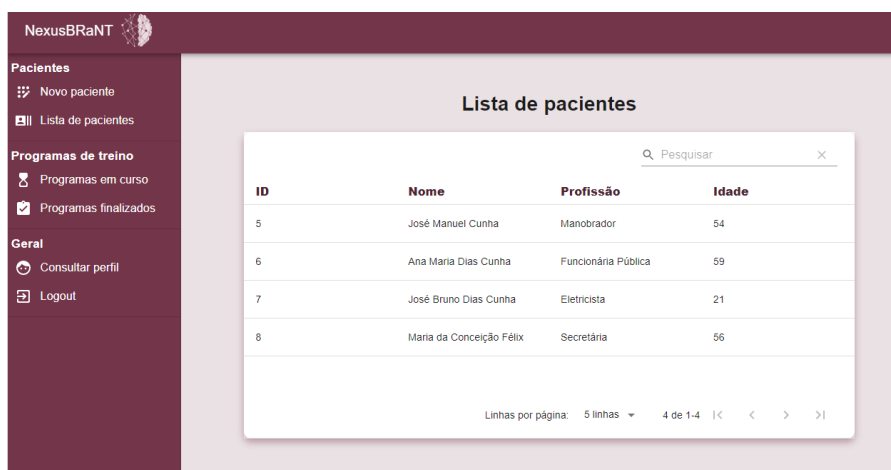


Fig. 4.24 – Funcionalidade de registo do paciente.

Observando a Fig. 4.24 verifica-se que todos os campos são obrigatórios para o registo de um novo paciente, ficando este associado ao PS autenticado. Caso sejam submetidos dados incorretos, como por exemplo data, contacto telefónico, email e números em vez de letras, o sistema irá avisar ao PS para corrigir esses campos de forma

a não serem guardados dados inválidos na base de dados. Este processo ocorre em todos os formulários desenvolvidos na plataforma, tal como referido na secção 4.3.6.

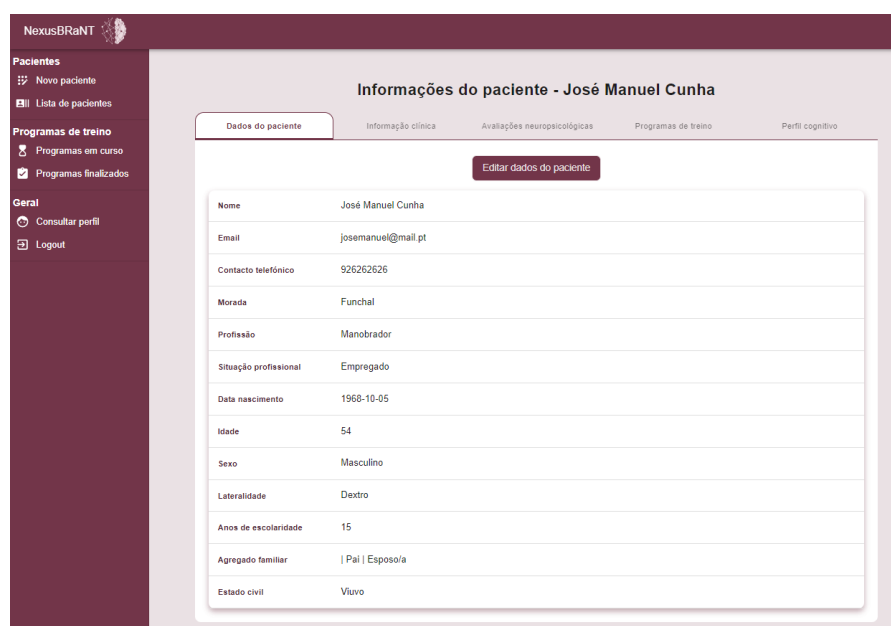
Na Fig. 4.25 está apresentada a listagem dos pacientes associados ao PS autenticado.



ID	Nome	Profissão	Idade
5	José Manuel Cunha	Manobrador	54
6	Ana Maria Dias Cunha	Funcionária Pública	59
7	José Bruno Dias Cunha	Eletricista	21
8	Maria da Conceição Félix	Secretária	56

Fig. 4.25 – Funcionalidade de listagem de pacientes associados ao PS autenticado.

O PS consegue pesquisar através da barra de pesquisa por um paciente, ordenar através do ID, do nome, da profissão ou da idade. Ao seleccionar o paciente desejado a plataforma reencaminha para a página de informações do mesmo, estando esta apresentada na Fig. 4.26.



Dados do paciente	
Nome	José Manuel Cunha
Email	josemanuel@mail.pt
Contacto telefónico	926262626
Morada	Funchal
Profissão	Manobrador
Situação profissional	Empregado
Data nascimento	1968-10-05
Idade	54
Sexo	Masculino
Lateralidade	Dextro
Anos de escolaridade	15
Agregado familiar	Pai Esposo/a
Estado civil	Viuvo

Fig. 4.26 – Funcionalidade de informações no paciente seleccionado.

É possível observar na Fig. 4.26 as informações sociodemográficas do paciente “José Manuel” no separador “Dados do paciente”, sendo que estas mesmas informações são possíveis de editar através do botão “Editar dados do paciente”. A funcionalidade de edição das informações do paciente utiliza o mesmo componente utilizado no registo do paciente, sendo esta apresentada na Fig. B.1 do anexo B. Tem-se nesta mesma página mais quatro separadores, sendo estes: informação clínica, avaliações neuropsicológicas,

programas de treino e perfil cognitivo. Na Fig. 4.27 apresenta o ecrã relativo à informação clínica do paciente.

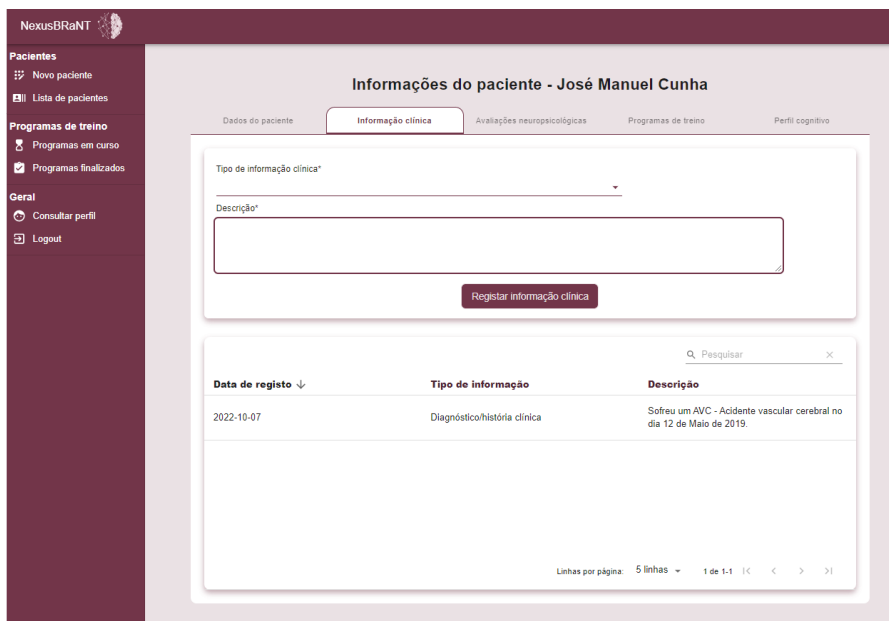


Fig. 4.27 – Funcionalidade da informação clínica do paciente selecionado.

Neste ecrã é possível registar uma informação clínica do paciente, selecionando o tipo de informação clínica (diagnóstico/história clínica, terapêutica farmacológica, exames complementares e registos clínicos das sessões) e descrever a informação. Na Fig. B.2 do anexo B está apresentada o exemplo da inserção de uma informação clínica. Após a inserção da informação clínica (ver Fig. 4.27) esta fica visível no componente abaixo do formulário (ver Fig. 4.27), que quando a mesma é selecionada é possível observar detalhadamente toda a informação clínica. É possível pesquisar e filtrar as informações clínicas através da data de registo, tipo de informação e descrição.

Na Fig. 4.28 está apresentada a informação clínica detalhada.

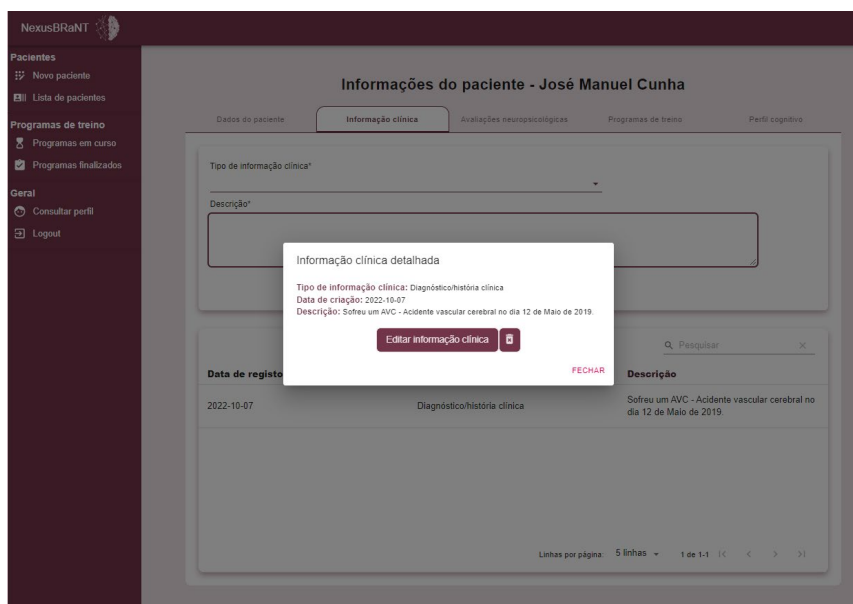


Fig. 4.28 – Funcionalidade da informação clínica detalhada.

Na Fig. 4.28 a informação clínica detalhada contém dois botões, editar e eliminar informação. Tal como foi referido anteriormente na secção 4.4.4, os formulários são componentes reutilizáveis, a edição da informação é semelhante ao seu registo, deste modo a mesma é apresentada na Fig. B.3 no anexo B. Na Fig. 4.29 é apresentada a funcionalidade de quando o PS deseja eliminar uma informação clínica. Esta funcionalidade é a mesma na eliminação de uma avaliação neuropsicológica e de um programa de treino cognitivo, sendo que posteriormente serão apresentadas ambas as funcionalidades.

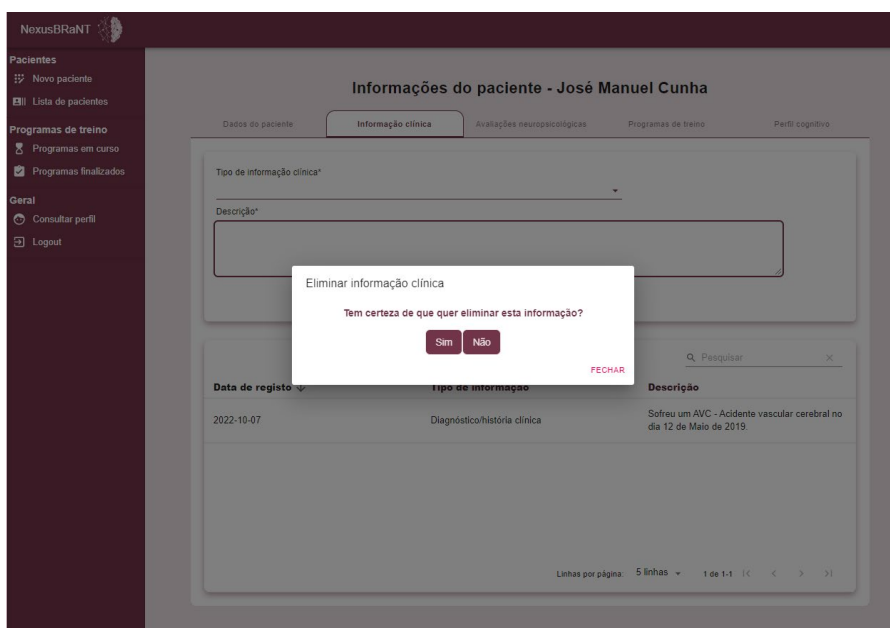


Fig. 4.29 – Funcionalidade de eliminar a informação clínica seleccionada.

Ao seleccionar o separador “Avaliações neuropsicológicas” na barra de navegação o seguinte ecrã é apresentado na Fig. 4.30.

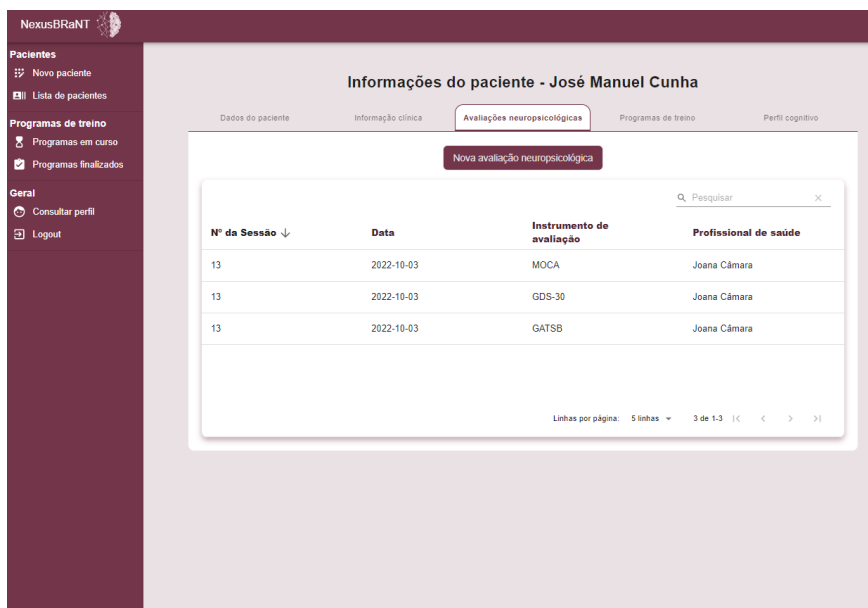


Fig. 4.30 – Funcionalidade das avaliações neuropsicológicas do paciente seleccionado.

É possível observar na Fig. 4.30 o botão que permite o registo de uma nova avaliação neuropsicológica e as avaliações neuropsicológicas já registadas. Ao selecionar o botão “nova avaliação neuropsicológica” é apresentada a funcionalidade da Fig. 4.31.

Fig. 4.31 – Funcionalidade de registo de uma nova avaliação neuropsicológica.

Na Fig. 4.31 está apresentada como exemplo o registo de uma avaliação neuropsicológica com o instrumento de avaliação “Toulouse-Piéron”. Os campos principais são os dados brutos da avaliação neuropsicológica, os subcampos estão relacionados com a interpretação clínica, sendo possível inserir a média e o desvio-padrão e seleccionar a interpretação clínica que se enquadra. De forma a facilitar qual a interpretação clínica que deverá ser seleccionada, foi disponibilizado no lado direito do título do instrumento de avaliação imagens com as tabelas dos dados normativos, estando apresentada na Fig. 4.32 a visualização das imagens referentes a teste instrumento.

Tabela de Índice de Dispersão

Table 7. Normative scores of the TP-DI according to age and education.

Age	Education			All education
	Primary (1-4)	middle (5-9)	High (>10)	
10-14				
n	32	37	43	112
M ± SD	21 ± 9	19 ± 12	12 ± 9	16 ± 10
SD	30, 35, 39	27, 31, 39	21, 26, 30	26, 31, 36
15-19				
n	36	24	39	99
M ± SD	26 ± 10	25 ± 8	21 ± 10	24 ± 9
SD	36, 41, 46	33, 37, 41	31, 37, 41	31, 38, 42
≥ 20				
n	85	42	19	146
M ± SD	28 ± 16	25 ± 14	25 ± 10	26 ± 13
SD	44, 53, 60	36, 41, 53	35, 40, 45	39, 46, 52
All age				
n	153	103	101	357
M ± SD	25 ± 12	22 ± 11	19 ± 10	22 ± 11
SD	37, 43, 49	33, 39, 44	29, 34, 39	33, 39, 44

Note: CI values are presented above 1 SD, 1.5 SD, and 2 SD, respectively.

(Lima et al., 2021)

Tabela de Índice de Eficiência

Table 6. Normative scores of the TP-WE according to age and education.

Age	Education			All education
	Primary (1-4)	middle (5-9)	High (>10)	
10-14				
n	32	37	43	112
M ± SD	160 ± 47	107 ± 46	191 ± 46	170 ± 50
SD	93, 70, 46	121, 98, 75	145, 122, 99	120, 95, 70
15-19				
n	36	24	39	99
M ± SD	92 ± 23	106 ± 24	147 ± 44	119 ± 30
SD	65, 52, 36	62, 70, 58	103, 79, 56	85, 69, 54
≥ 20				
n	85	42	19	146
M ± SD	89 ± 20	99 ± 25	135 ± 41	100 ± 29
SD	60, 50, 40	70, 57, 45	84, 63, 43	71, 56, 41
All age				
n	153	103	101	357
M ± SD	106 ± 20	122 ± 32	154 ± 44	138 ± 36
SD	74, 59, 44	91, 74, 58	110, 88, 66	92, 74, 56

Note: WE values are presented below 1 SD, 1.5 SD, and 2 SD, respectively.

(Lima et al., 2021)

Fig. 4.32 – Funcionalidade de visualização dos dados normativos do instrumento "Toulouse-Piéron".

Consoante o instrumento de avaliação selecionado, este pode conter interpretação clínica ou não. A interpretação clínica divide-se em cognitiva e não cognitiva. Na Fig. B.4 a) e b) do anexo B são apresentados quais instrumentos de avaliação contém interpretação clínica e qual categoria pertencem, como também as escalas presentes nas opções do campo de seleção. Estas informações foram fornecidas pelas PS que auxiliaram na elicitação dos requisitos.

Na visualização das avaliações cognitivas já registadas é possível pesquisar e ordenar as mesmas através do ID da sessão, data, instrumento de avaliação e profissional de saúde. Na Fig. 4.30 é possível visualizar as avaliações registadas e ao selecionar-se o instrumento desejado para obter todas as informações detalhadas da avaliação neuropsicológica tem-se o ecrã da Fig. 4.33.

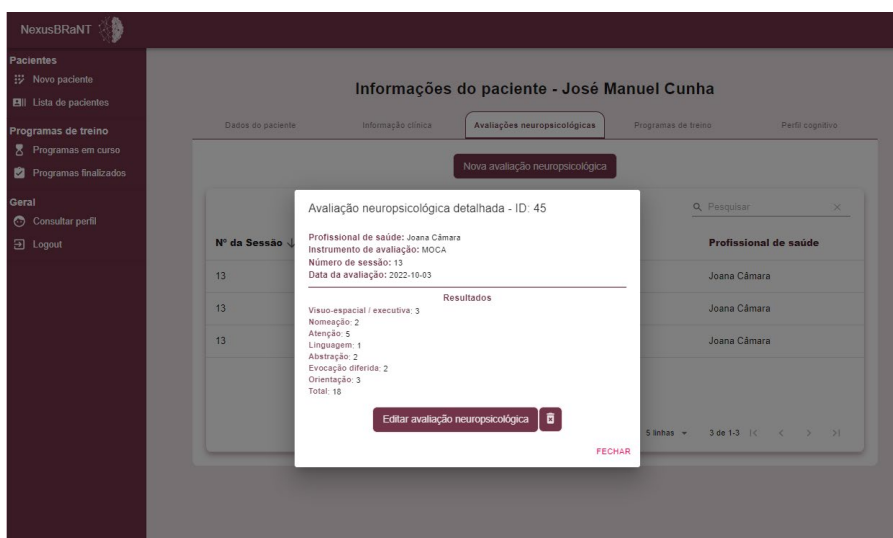


Fig. 4.33 – Funcionalidade da avaliação neuropsicológica detalhada selecionada.

Tal como nos dados do paciente e na informação clínica, a edição da avaliação neuropsicológica tem formulário semelhante ao seu registo. Na Fig. B.5 do anexo B é possível observar a edição da avaliação neuropsicológica selecionada na Fig. 4.33. A funcionalidade de eliminar a avaliação neuropsicológica selecionada é mesma que foi apresentada na Fig. 4.31, na Fig. B.6 do anexo B é apresentada a correspondente funcionalidade de eliminação da avaliação neuropsicológica selecionada.

O penúltimo separador é “Programas de treino” que contém o acesso ao formulário de registo de um programa de treino e visualização dos programas do paciente. Na Fig. 4.34 está apresentado o ecrã das informações dos programas de treinos associados ao paciente selecionado.

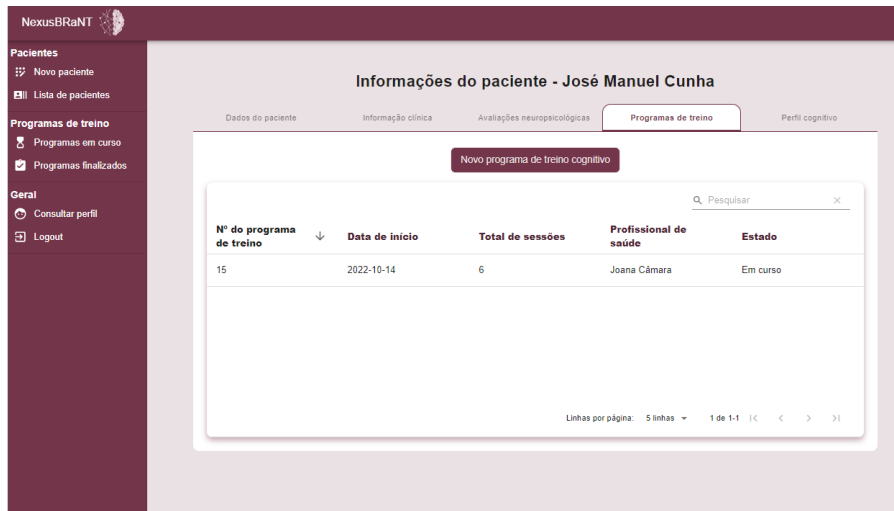


Fig. 4.34 – Funcionalidade dos programas de treino do paciente seleccionado.

Através do botão “Novo programa de treino cognitivo” a plataforma encaminha o PS para o registo de um novo programa de treino cognitivo, sendo este apresentado na Fig. 4.35. O PS regista a data que deseja que o paciente realize a primeira sessão de treino, o número total de sessões, o número de vezes por semana que as mesmas deverão ser realizadas, a duração de cada sessão e o desafio para adaptação.

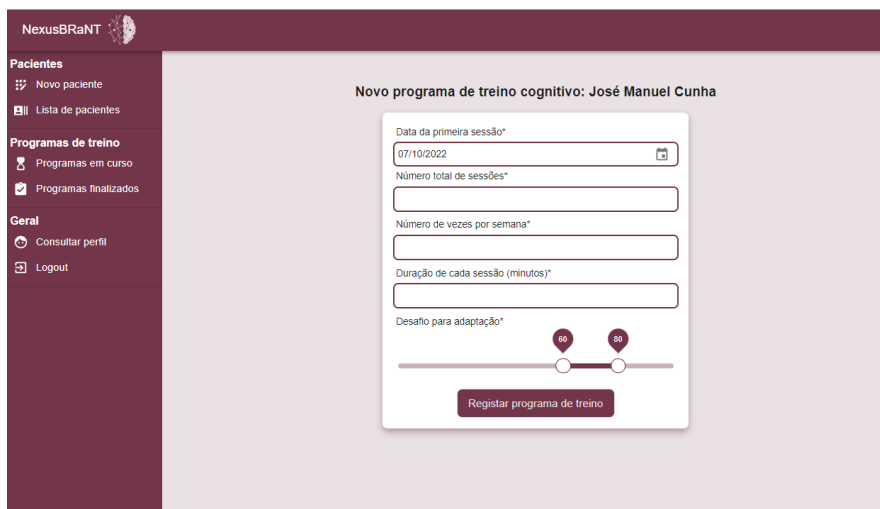


Fig. 4.35 – Funcionalidade de registo de um programa de treino cognitivo.

É possível pesquisar e ordenar os programas de treino através do ID do programa de treino, data de início, total de sessões e profissional de saúde. Quando o PS selecciona um programa de treino é apresentada a informação detalhada do mesmo, tal como apresentado na Fig. 4.36.

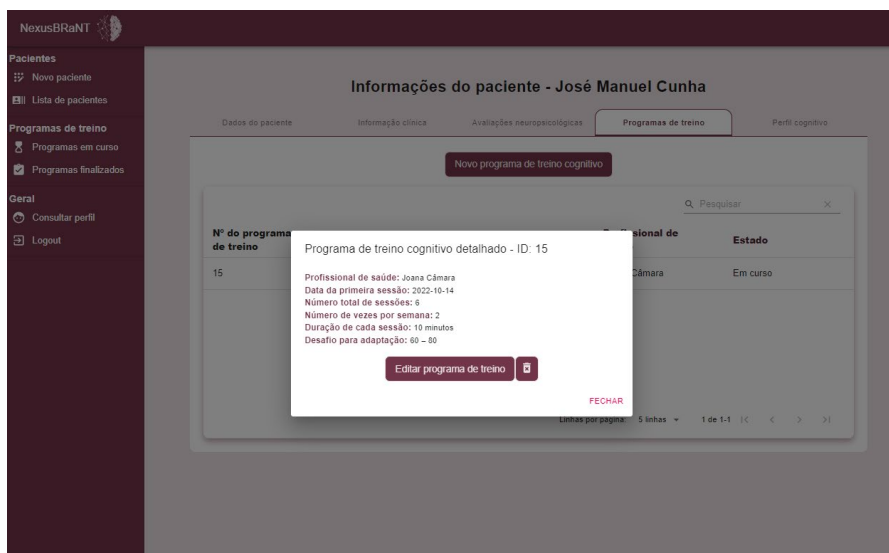


Fig. 4.36 – Funcionalidade do programa de treino detalhado selecionado.

É possível editar o programa de treino e até eliminá-lo, estando ambos dois estão apresentados nas Fig. B.8 e Fig. B.7 no anexo B, respetivamente. A edição do programa de treino utiliza o formulário de inserção do mesmo.

Por fim no último tem-se o separador “Perfil cognitivo” (ver Fig. 4.37).

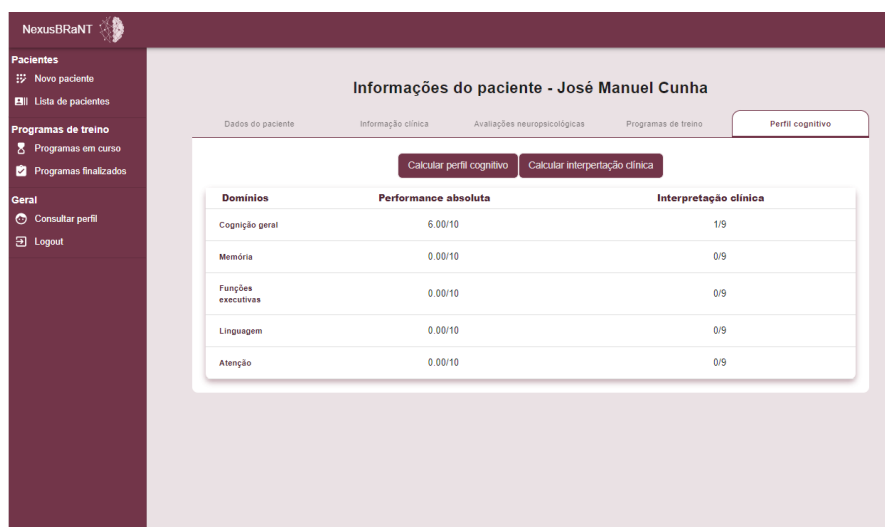


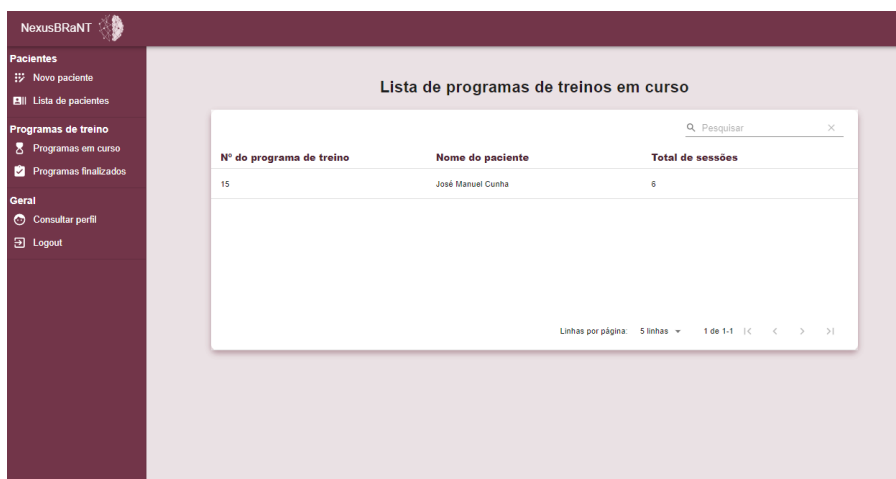
Fig. 4.37 – Funcionalidade do perfil cognitivo.

A Fig. 4.37 apresenta a interface responsável pela visualização da performance absoluta e da interpretação clínica nos diversos domínios apresentados, sendo estes a cognição geral, a memória, as funções executivas, a linguagem e a atenção. Ao clicar no botão “Calcular perfil cognitivo” ou no botão “Calcular interpretação clínica” são calculados os mesmos através de um algoritmo implementado por outro elemento do projeto, sempre tendo em conta todas as avaliações neuropsicológicas associadas a última de sessão de avaliações.

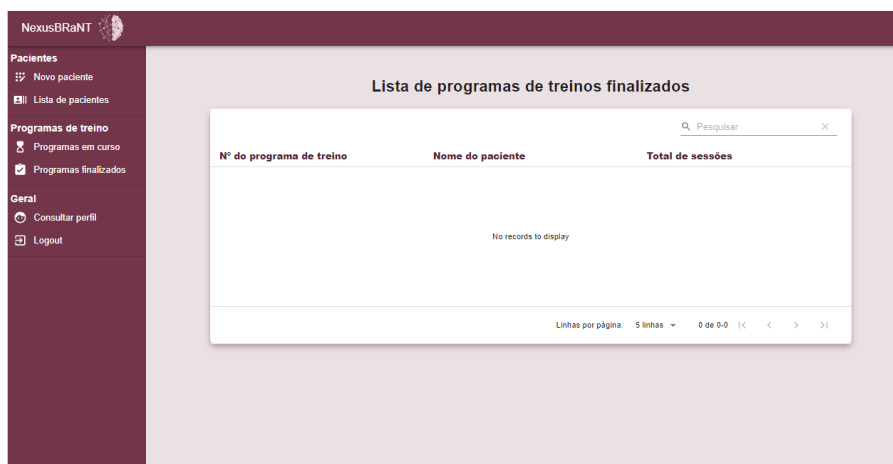
A categoria dos programas de treino, no menu dos PS clínicos e investigadores (Fig. 4.21 a)), contém duas subcategorias: programas em curso e programas finalizados. Em

ambas as funcionalidades tem-se uma lista dos programas correspondentes a cada um, no qual é possível selecionar o programa, e a plataforma reencaminha o PS para a visualização detalhada dos programas de treino, sessões e progresso das mesmas.

A Fig. 4.38 a) e b) apresenta a listagem dos programas de treino em curso e a listagem dos programas de treino finalizados.



a)



b)

Fig. 4.38 – Funcionalidade de listas de programa de treinos: a) em curso; b) finalizados.

É utilizado o mesmo componente para a listagem dos programas de treino e para a visualização detalhada referida anteriormente. Os programas de treino poderão estar concluídos ou a decorrer, para tal é passada uma variável indicando o estado do programa de treino, e assim, através da mesma solicita-se a API os programas de treino desejados.

Na Fig. 4.39 é apresentado um programa de treino em curso, no qual é possível verificar todas as sessões, as suas datas de início e fim e o estado das mesmas. Pode-se pesquisar ou ordenar as sessões de treino através do número de sessão, data de início, data de fim e estado.

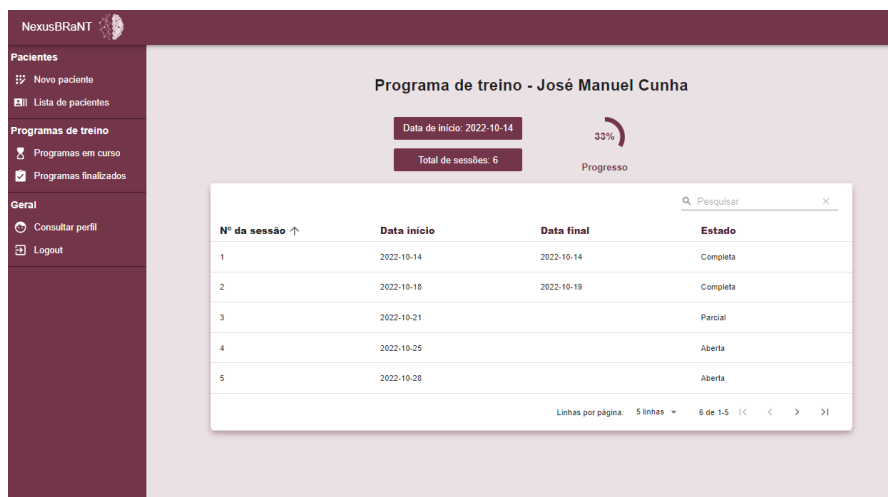


Fig. 4.39 – Funcionalidade de detalhe do programa de treino em curso selecionado.

Sempre que uma sessão de treino é concluída, o seu estado é alterado e a barra de progresso é alterada conforme a percentagem de sessões que são concluídas. Ao clicar numa sessão é possível ver em detalhe todas as suas informações, tais como: data de início, data de fim, duração total, estado, performance, notas e parâmetros de intervenção. Todas estas informações provêm do sistema externo dos jogos, nos quais o paciente realizou remotamente, sendo este desenvolvido por outros elementos do projeto.

Na Fig. 4.40 é possível visualizar o ecrã de uma sessão de treino detalhada.

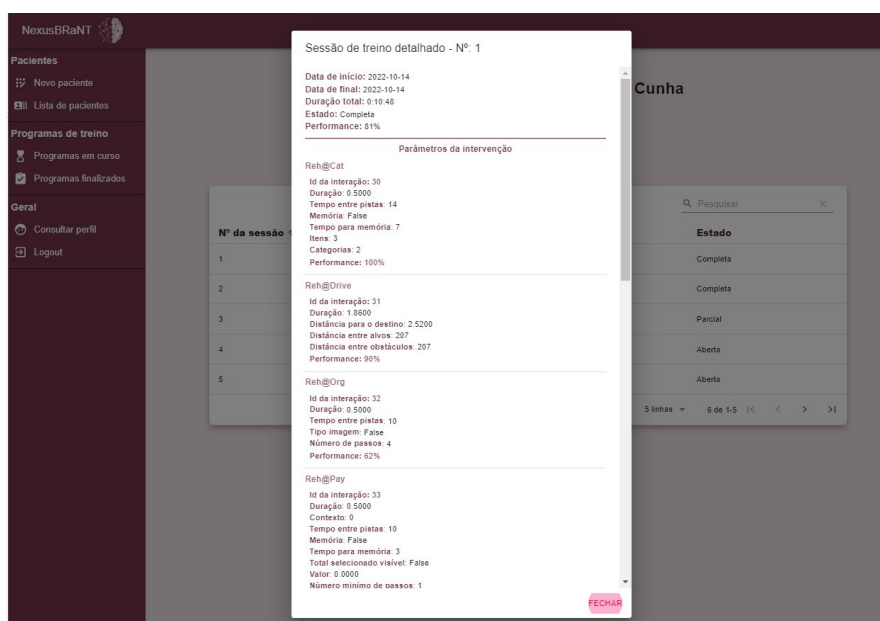


Fig. 4.40 – Funcionalidade da visualização detalhada da sessão de treino detalhada.

Tal como referido anteriormente, para os programas de treino finalizados ocorrem as mesmas iterações, sendo estas: seleção do programa de treino, visualização de todas as sessões e seleção da sessão em detalhe.

Por último tem-se no menu (Fig. 4.21 a)) a categoria geral que contém a funcionalidade de consultar o perfil do PS autenticado e a funcionalidade de *logout*. Na Fig. 4.41 é exibida a funcionalidade de perfil do PS autenticado na plataforma.



Fig. 4.41 – Funcionalidade de visualização dos dados do utilizador autenticado.

É possível observar na Fig. 4.41 os dados do PS, sendo estes: nome, email, cédula profissional e contacto telefónico. Estes quatro dados podem ser editados pelo próprio PS e na Fig. 4.42 está apresentado essa funcionalidade. Todos os campos são obrigatórios, sendo importante destacar a cédula profissional, pois foi necessário garantir que o PS registado na plataforma tenha cédula oficial. Devido ao facto de tratar-se de dados protegidos sobre a saúde de um paciente, é essencial garantir o profissionalismo do PS.

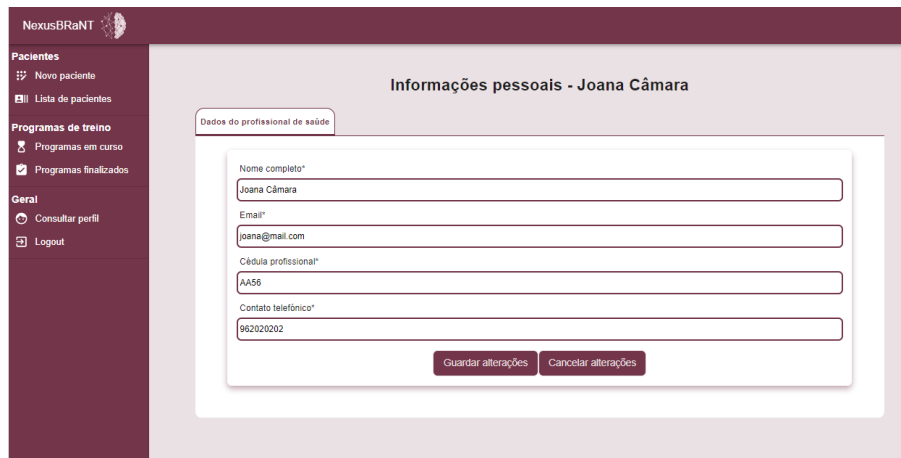


Fig. 4.42 – Funcionalidade de edição dos dados do PS autenticado.

Tal como em todos os formulários, a edição dos dados do PS é semelhante ao registo do mesmo, o qual será apresentado posteriormente. O PS ao realizar o *logout* na plataforma é encaminhado para a página de autenticação novamente.

4.4.5.2. Administrador

O administrador do sistema tem como funcionalidade principal a gestão de utilizadores e de si mesmo. Na categoria de utilizadores apresentada no menu (Fig. 4.21 b)) tem-se duas subcategorias: registo de um novo utilizador e lista de utilizadores.

Na Fig. 4.43 está apresentada a funcionalidade de registo de um profissional de saúde.

The screenshot shows a web interface for registering a health professional. On the left is a dark sidebar with the 'NexusBRaNT' logo and navigation options: 'Utilizadores' (with sub-items 'Novo utilizador' and 'Lista de utilizadores'), and 'Geral' (with sub-items 'Consultar perfil' and 'Logout'). The main content area is titled 'Registrar profissional saúde' and contains a form with the following fields: 'Nome completo*' (empty), 'Email*' (filled with 'maria@teste.com'), 'Tipo de conta*' (dropdown menu), 'Cédula profissional*' (empty), 'Contacto telefónico*' (empty), 'Password*' (filled with '*****'), and 'Confirmar password*' (empty). A 'Registrar utilizador' button is located below the form.

Fig. 4.43 – Funcionalidade de registo de um utilizador.

Na Fig. 4.43 tem-se vários campos, entre os quais: nome completo, email, tipo de conta, cédula profissional, contacto telefónico, *password* e confirmar *password*. Estes campos são obrigatórios a serem preenchidos para que o registo do profissional de saúde seja bem-sucedido.

Quando o administrador acede à lista de utilizadores é possível pesquisar ou ordená-los através do seu ID, nome, cédula profissional ou contacto telefónico, estando esta funcionalidade apresentada na Fig. 4.44.

The screenshot shows a web interface for listing health professionals. On the left is the same dark sidebar as in Fig. 4.43. The main content area is titled 'Lista de profissionais de saúde' and features a search bar with the text 'Pesquisar' and a close button. Below the search bar is a table with the following data:

ID	Nome	Cédula profissional	Contacto telefónico
1	Joana Câmara	AA56	962020202
2	Érica Filipa Dias Cunha	123	969696965

At the bottom of the table, there are pagination controls: 'Linhas por página: 5 linhas', '2 de 1-2', and navigation arrows.

Fig. 4.44 – Funcionalidade de listagem dos profissionais de saúde.

Ao ser seleccionado um profissional de saúde, a plataforma encaminha o PS para a página dos dados do PS seleccionado. O administrador tem acesso aos dados e pode editar os mesmos sendo que a página de edição está apresentada na Fig. B.9 do anexo B. É possível editar também o tipo de conta do PS seleccionado através do mesmo formulário.

Na Fig. 4.45 é apresentada a funcionalidade de visualização de dados do PS, após a sua seleção.



Fig. 4.45 – Funcionalidade de visualização de dados do profissional de saúde selecionado.

Na barra de navegação existem outras duas subcategorias associadas ao PS selecionado anteriormente, sendo estas: pacientes associados e associar/desassociar paciente. Ao selecionar os pacientes associados, o administrador tem acesso aos mesmos, mas só apenas ao seu ID no sistema, nome e email, devido à proteção de dados. Esta funcionalidade está apresentada na Fig. 4.46.

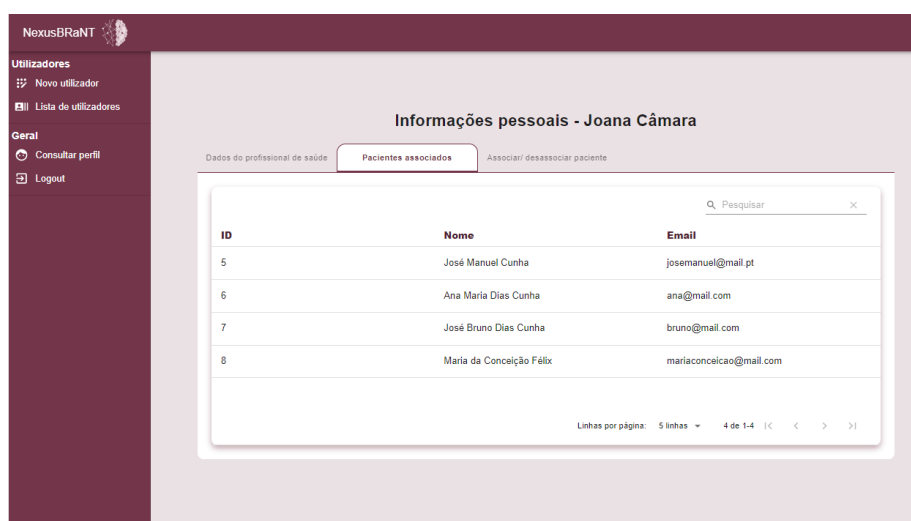


Fig. 4.46 – Funcionalidade de listagem dos pacientes associados do PS selecionado.

Na barra de associar/desassociar paciente é possível obter os pacientes que estão associados ao PS selecionado, dos quais é possível desassociar do PS. Todos aqueles que não estão associados a nenhum PS são obtidos e apresentados na seleção de associar paciente ao PS selecionado.

Na Fig. 4.47 apresenta o ecrã de associar/desassociar paciente.



Fig. 4.47 – Funcionalidade de associar/desassociar paciente.

A Fig. B.10 a) b) do anexo B apresenta uma iteração de desassociação de um paciente e o mesmo posteriormente aparece no campo de associar paciente, sendo possível associar o mesmo novamente. Criou-se esta funcionalidade de modo a ser possível associar/desassociar paciente quando um PS ficar ausente da plataforma. Assim sendo, é possível outro PS continuar o acompanhamento do paciente.

Tal como nos utilizadores comuns, o menu de administrador (Fig. 4.21) tem a categoria geral, onde é possível consultar o perfil do administrador autenticado e editar os seus próprios dados. Na Fig. 4.48 está apresentado a funcionalidade de visualização de dados do administrador.



Fig. 4.48 – Funcionalidade da visualização dos dados do administrador autenticado.

Na Fig. B.1 do anexo B é apresentada a funcionalidade da edição dos dados do administrador autenticado na plataforma. A funcionalidade de *logout* é exatamente igual à apresentada anteriormente, sendo que esta permite voltar para a página de início de sessão quando o PS seleciona o botão *logout*. Na Fig. B.12 do anexo B está apresentado o ecrã quando o endereço da plataforma está incorreto, indicando assim que página acedida não foi encontrada, tanto no lado dos PS clínicos ou investigadores como no lado do administrador.

4.5. Conclusões

Durante todo o desenvolvimento da plataforma e API foi realizada uma avaliação contínua aos mesmos. Foram realizados testes funcionais para cada funcionalidade da plataforma e testes funcionais para cada rota criada na API. Também foram realizados testes de integração entre a plataforma desenvolvida e API e entre a API e o sistema externo (Reh@Sync). Quando a funcionalidade da plataforma e a rota na API eram executadas com sucesso nos testes funcionais iniciava-se os testes de integração entre ambas, sem ocorrência de erros retornava-se ao desenvolvimento das funcionalidades e assim sucessivamente.

O método de desenvolvimento escolhido neste projeto foi essencial para a implementação desta tese, pois pode-se manter contacto contínuo com os elementos de projeto. Deste modo, obteve-se o *feedback* em relação todas as funcionalidades desenvolvidas e foram realizadas as alterações necessárias consoantes as solicitações dos mesmos. As funcionalidades desenvolvidas apresentadas neste capítulo fizeram parte do leque dos requisitos considerados mais importantes para o projeto de investigação, sendo estes priorizados forma iterativa.

5. Avaliação

Neste capítulo são apresentados os testes realizados ao sistema implementado e a avaliação ao mesmo.

Primeiramente, ao longo da implementação do sistema, foi necessário realizar testes formais e informais com as PS de todas as funcionalidades individualmente, de forma a avaliar se as mesmas estavam dentro do pretendido.

Após o sistema estar completamente implementado foram realizados alguns testes de usabilidade, sendo estes: o questionário *System Usability Scale* (SUS), *think aloud* e análise da utilização da interface. Realizou-se os testes aos utilizadores finais, que neste sistema são PS que trabalham na área e alunos do mestrado de psicologia.

Os critérios utilizados para avaliação do sistema serão apresentados posteriormente, mas estes foram definidos por elementos do projeto BRaNT. As *experts* do sistema que ajudaram na elicitação também realizaram *user testing* de forma a realizarem a sua avaliação final ao sistema.

5.1. Procedimento dos testes de usabilidade

Os testes de usabilidade dividiram-se em 6 fases, sendo estas: apresentação, ambientação com o NexusBRaNT, tarefas, questionário SUS, *feedback* final e conclusão. Na realização dos testes, foram escolhidas as tarefas mais relevantes no projeto, tendo em conta pelo menos uma tarefa de cada grupo principal das funcionalidades (gestão de pacientes, gestão de programa de treinos e gestão de avaliações neuropsicológicas) abordado na secção 1.3. O método *think aloud* é um protocolo usado para obter os dados e *feedback* nos testes dos utilizadores. Os mesmos, ao testar a plataforma, indicam em voz alta os aspetos positivos e negativos da mesma, permitindo assim uma melhor perceção dos participantes com o sistema testado [53]. Todos os testes foram realizados individualmente e presencialmente, numa sala reservada, sem influência externas.

Na Tabela 5.1 estão apresentadas as fases realizadas nos testes de usabilidade e a duração média de cada atividade realizada. Cada teste de usabilidade teve a duração média total de 30 a 40 minutos.

Tabela 5.1 – Procedimentos dos testes de usabilidade.

Fases		Duração média (minutos)
Apresentação	Introdução - BRaNT	3
	NexusBRaNT	2
Ambientação com a plataforma	Exploração livre	3-5
Tarefas	1. Novo paciente	2
	2. Nova avaliação neuropsicológica	3
	3. Calcular perfil cognitivo	1
	4. Novo programa de treino	2
	5. Visualização das performances numa sessão de treino	2
Questionário - SUS	<i>Google Forms</i>	3-5
<i>Feedback</i> final	Funcional/ <i>design</i>	3-5
Conclusão	Apresentação da plataforma como administrador	3
	Agradecimento	1
		Total
		30-40

O protocolo do teste de usabilidade seguido com todos os participantes foi o seguinte (apresentado na Tabela 5.1):

- **Apresentação:** foi criado um *PowerPoint* para que os participantes tenham uma ligeira percepção do porquê da implementação do sistema, onde se apresentou o projeto de investigação BRaNT, a ferramenta NeuroAIRh@b e a plataforma NexusBRaNT. Explicou-se o que era plataforma desenvolvida, qual o seu público-alvo e as principais funcionalidades da mesma. Na Fig. C.1 a) b) c) d) e) do anexo C são apresentados os *slides* apresentados aos participantes. Apresentou-se na plataforma as funcionalidades principais que foram previamente apresentadas nos *slides*. Depois prosseguiu-se com a explicação das próximas fases que o participante iria realizar.
- **Ambientação com a plataforma:** cedeu-se o computador para que o participante pudesse testar a plataforma. O *site* e a API já estavam a ser executados localmente no computador. Ambos ainda não estavam *online* pois um estudo piloto (no âmbito do projeto de investigação) ocorria em paralelo com uma versão da plataforma que não continha ainda a interpretação clínica nem as funcionalidades de eliminação. O participante teve a liberdade de utilizar a plataforma, explorando todas as funcionalidades existentes. Durante a sua ambientação com a plataforma o participante foi dando *feedback* da mesma, usando-se assim o método *think aloud*.
- **Tarefas:** posteriormente à exploração do *site* começou-se a indicar as tarefas a serem realizadas sequencialmente após a conclusão de cada uma delas. Não se definiu dados a serem inseridos para que o participante tivesse liberdade de escolha e também para que os dados inseridos não fossem “viciados”, provando desta forma que a plataforma estava preparada para inserção de diversos dados distintos. Tal como a fase anterior, utilizou-se assim o método *think aloud*. As tarefas indicadas para os participantes realizarem são as seguintes:
 1. **Novo paciente:** foi solicitado ao participante para inserir um novo paciente para que o mesmo pudesse utilizar o paciente inserido para executar as tarefas seguintes.
 2. **Nova avaliação neuropsicológica:** solicitou-se ao participante para inserir uma nova avaliação à sua escolha. Quando o participante inseria uma avaliação não cognitiva, foi solicitado novamente ao participante para inserir uma avaliação neuropsicológica só para ser possível visualizar a interação realizada na tarefa seguinte.
 3. **Calcular perfil cognitivo:** o cálculo do perfil cognitivo provém dos resultados brutos das avaliações neuropsicológicas inseridas na última de sessão de avaliações neuropsicológicas. Deste modo solicitou-se ao participante para calcular o perfil cognitivo e indicar quais os domínios cognitivos que foram alterados após o cálculo.
 4. **Novo programa de treino:** esta tarefa tinha como objetivo a inserção de um programa de treino cognitivo no paciente registado pelo participante. Ainda na sequência desta tarefa foi pedido para que o participante editasse o desafio para adaptação (para 20-50) no programa de treino inserido.
 5. **Visualização das performances numa sessão de treino:** na plataforma já estavam alguns dados inseridos para que fosse possível o participante explorar. Desta forma, um programa de treino de um paciente inserido (antes

dos testes de usabilidade) já continha sessões abertas, parciais e concluídas. Então foi pedido para que o participante visualizasse uma sessão de treino do paciente indicado e verificasse as performances das atividades cognitivas que o paciente realizou.

- **Questionário – SUS:** após a realização de todas as tarefas, o participante preencheu o questionário SUS acessado através do *Google Forms*. Dividiu-se o formulário em 4 secções: informações acerca do estudo a ser realizado (objetivo, procedimento, critério de inclusão, riscos, benefícios, compensação e custos e a confidencialidade), consentimento e autorização (direitos, questões, contactos e consentimento voluntário informado), dados demográficos do paciente (*user ID*, idade, nacionalidade, género e estado profissional) e por fim as questões do SUS. Na Fig. C.2 a) b) c) d) do anexo C é possível visualizar o questionário do *Google Forms* realizado pelos participantes.
- **Feedback final:** solicitou-se ao participante para indicar qual era o seu *feedback* final em relação às funcionalidades e ao *design*.
- **Conclusão:** foi apresentada a plataforma do lado do administrador (novo utilizador, lista de utilizadores, dados e pacientes associados do utilizador selecionado e associar/desassociar paciente). Por fim concluiu-se o teste de usabilidade com um agradecimento na participação do estudo.

5.2. Participantes

Nesta subsecção serão apresentados os participantes que realizaram o teste de usabilidade descrito anteriormente. Neste estudo participaram 16 participantes com idades entre os 21 e 48 anos. O critério de inclusão para o estudo foi ser profissional de saúde ou estudante de saúde, mais precisamente na área da psicologia ou neuropsicologia. Na Tabela 5.2 é apresentado um resumo dos dados demográficos dos participantes.

Tabela 5.2 - Dados demográficos dos participantes.

ID do participante	Idade	Nacionalidade	Género	Estado profissional
1	37	Portuguesa	Feminino	Profissional de saúde no BRaNT
2	25	Portuguesa	Feminino	Estudante de mestrado
3	29	Portuguesa	Feminino	Profissional de saúde
4	27	Portuguesa	Feminino	Profissional de saúde
5	21	Portuguesa	Feminino	Estudante de mestrado
6	26	Portuguesa	Masculino	Estudante de mestrado
7	26	Portuguesa	Feminino	Estudante de mestrado
8	21	Portuguesa	Feminino	Estudante de mestrado
9	21	Portuguesa	Feminino	Estudante de mestrado
10	35	Portuguesa	Feminino	Profissional de saúde
11	25	Portuguesa	Feminino	Profissional de saúde
12	34	Portuguesa	Feminino	Profissional de saúde
13	28	Portuguesa	Feminino	Profissional de saúde
14	48	Portuguesa	Feminino	Profissional de saúde
15	36	Portuguesa	Feminino	Profissional de saúde
16	29	Portuguesa	Feminino	Profissional de saúde no BRaNT

Observando a Tabela 5.2 é possível verificar que 8 dos participantes foram PS, 6 estudantes de mestrado e 2 PS que são do projeto do BRaNT, sendo estas as *experts* no sistema.

5.3. Avaliação ao sistema

Tendo em conta os três tipos de participantes decidiu-se dividir a avaliação do sistema em 3 grupos: estudantes de mestrado, PS e *experts* do BRaNT. A avaliação qualitativa será separada consoante os grupos e a quantitativa apresentada com todos os participantes.

5.3.1. Avaliação qualitativa

Nesta subsecção será apresentada toda a avaliação qualitativa proveniente dos testes de usabilidade realizados. Como descrito acima esta avaliação foi dividida pelos três grupos de utilizadores tendo em conta as diferentes características dos mesmos, como por exemplo: anos de experiência, experiência profissional, prática com os instrumentos de avaliação neuropsicológica e conhecimento/experiência com a plataforma. Decidiu-se dividir a avaliação por funcionalidade e não por participante, devido ao facto de algumas das opiniões serem comuns.

5.3.1.1. Estudantes de mestrado

Os estudantes de mestrado que foram recrutados para os testes de usabilidade são 5 alunos da Universidade da Madeira e 1 aluno da Universidade de Coimbra, que atualmente se encontra na região. Enquanto os alunos exploravam o *site* foram questionando o que podiam ou não aceder, sendo que foi dada total liberdade de acederem às funcionalidades desejadas de forma que os mesmos ficassem confortáveis ao explorar.

No registo de um paciente, o campo mais questionado foi o da escolaridade, devido a forma em que a mesma é apresentada (de zero a vinte), no qual se torna confuso a escolha de uma licenciatura, mestrado ou doutoramento através de valores numéricos e não por nível académico. Também foi referido, no campo de situação profissional, que deveria ser inserido mais uma seleção que equivale à “invalidéz” do paciente, pois só existem os campos “empregado”, “desempregado”, “de baixa” e “reformado”.

Após a realização da tarefa 2 (avaliação neuropsicológica), alguns dos alunos referiram que a plataforma tem a vantagem de ter uma vasta escolha de instrumentos de avaliação. A razão pela qual existem uma grande variedade de instrumentos é devido ao facto que, quando é realizada uma avaliação a um paciente, são utilizados diversos instrumentos numa só avaliação.

Na funcionalidade do perfil cognitivo do paciente (tarefa 3), alguns dos participantes indicaram que o facto de ser possível calcular o mesmo através de o pressionar de um botão permite tornar o processo mais eficiente e mais rápido em vez de o PS efetuar este mesmo cálculo de forma manual.

Na última tarefa realizada pelos participantes foi possível visualizar o progresso do paciente num dos campos do *site* (apresentado de forma gráfica e em percentagem), algo que os mesmos indicaram ser bastante vantajoso, pois permite o PS facilmente visualizar quanto é que falta para o paciente completar o programa de treino cognitivo.

Após a conclusão das tarefas os estudantes de mestrado deram o *feedback* geral da plataforma incluindo o *design* da mesma. De um modo geral, os participantes indicaram que a plataforma era de fácil utilização, acessível a qualquer utilizador, perceptível e a navegação da mesma era bastante intuitiva. O facto de os dados do paciente estarem organizados por categorias foi uma das vantagens indicadas, pois deste modo e, apesar de ser a primeira interação e pouca experiência na área, conseguiram executar todas as tarefas com sucesso.

Um aspeto importante a referir está relacionado a ver com uma experiência de estágio de verão que um participante (estudante de psicologia) teve. Neste estágio ele confidenciou que teve de utilizar a plataforma *Excel* e uma base de dados para inserir os dados do paciente em ambos, sendo que a base de dados era utilizada por todos os PS e não era focada nos psicólogos, tornando esta situação numa desvantagem. O participante realizava a inserção no *Excel* e na base de dados, visto que o processo de visualizar as informações do paciente na base de dados era complexo e demorado, sendo que o *Excel* era mais rápido a visualização e mais intuitivo.

5.3.1.2. Profissionais de saúde

Os PS que foram escolhidos para a realização dos testes de usabilidade já tinham experiência na área e trabalham em contextos diferentes, sendo estes: clínica privada, serviço de psicologia da Universidade da Madeira e docentes. Tal como os estudantes de mestrado, os PS tiveram total liberdade na ambientação da plataforma.

Tal como nos participantes anteriores começou-se pela tarefa de registo de um paciente no qual foram indicados algumas observações positivas e alguns aspetos que poderiam ser adicionados de forma a complementar ainda mais a plataforma. Um dos pontos referidos foi o facto dos campos de seleção já terem informação que poderá ser selecionada, facilitando essa escolha aos PS. Um dos participantes tentou registar o paciente com uma data de nascimento inválida e a plataforma notificou o mesmo, sendo que deveria inserir uma data correta e esta notificação foi um ponto positivo indicado, pois não deixou o PS avançar. Um dos pontos referidos pelos outros participantes acima sobre a seleção do nível de escolaridade foi também indicado pelos PS, visto que a seleção do nível de forma numérica torna-se um pouco confuso. Na seleção múltipla do agregado familiar os campos selecionados deveriam estar com pouco mais de destaque na cor, sendo um ponto referido pelos participantes. O último ponto referido por dois participantes era a existência da inserção do contacto de emergência de um cuidador ou de um acompanhante, pois, apesar de não influenciar nos dados do paciente, este contacto poderia ser necessário para estabelecer contacto quando o programa de treino está ocorrendo remotamente e o paciente necessita de algum suporte por parte de um familiar/cuidador.

Um dos pontos que apresentaram um realce mais positivo, nas avaliações neuropsicológicas, foi o facto da visualização dos dados normativos dos instrumentos de avaliação, através das imagens cedidas e inserção desses dados, e da interpretação clínica em cada instrumento de avaliação. Estes dados atualmente são consultados em papel, fazendo que esta consulta demore algum tempo. A inclusão destes dados no *site*, juntamente com a inserção dos resultados do instrumento facilita a análise, melhorando eficiência dos PS. Um outro ponto positivo que foi referido pelos participantes foi a

diversidade de instrumentos de avaliação e a descrição do mínimo e máximo dos seus campos.

Na tarefa de calcular o perfil cognitivo ocorreram o mesmo tipo de observações, ou seja, que o facto de ser possível calcular automaticamente o perfil cognitivo e a interpretação clínica ajudam os PS neste processo, pois o mesmo tende a ser realizado manualmente e necessitam de consultar todos os resultados das avaliações para executar esse cálculo.

Nas tarefas de criação de um programa de treino e visualização das sessões de treino foram indicados alguns aspetos acerca da falta de informação sobre em que domínios as atividades cognitivas realizadas pelos pacientes se enquadravam. Deste modo, alguns participantes sugeriram um resumo das atividades cognitivas e dos domínios predominantes juntamente com os dados dos programas e sessões de treino, tendo assim uma informação mais qualitativa para além da performance (informação quantitativa). Sendo que visualização da performance foi um ponto positivo referido pelos mesmos.

Apesar das tarefas não englobarem as funcionalidades da informação clínica, alguns dos participantes na ambientação com a plataforma exploraram estas e referiram que os vários tipos de informações clínicas possíveis de ser inseridas englobam diversas informações que utilizam profissionalmente.

Concluindo as tarefas que tinham de executar na plataforma foi indicado de um modo geral aspetos positivos e aspetos que poderiam ser adicionados/melhorados na plataforma de forma que a mesma ficasse mais completa. A plataforma foi referida como muito intuitiva, agradável e apelativa, devido a existir um menu à esquerda com as principais funcionalidades, e a barra de navegação das informações do paciente que organizam, facilitando a visualização dos dados disponíveis. O *design* da plataforma foi considerado minimalista, tendo cores suaves e sem excesso de informações. Um dos pontos referidos como em falta foi existir um administrador por instituição e não um administrador da plataforma inteira. Deste modo o administrador apenas poderia visualizar a lista de pacientes e PS da sua instituição, sendo este ponto referido se a aplicação fosse distribuída no futuro. Por último, dois participantes referiram que este tipo de plataformas serão o futuro na área da saúde e, apesar de ser a primeira interação com a plataforma, conseguiram realizar todas as tarefas sem o mínimo de esforço necessário, sendo que este último ponto foi referido não só por estes dois participantes, mas por todos os participantes deste grupo.

Tal como ocorreu no grupo de participantes anterior, um participante deste grupo indicou que na altura que trabalhou num hospital público e a plataforma disponibilizada pelo mesmo era mais complexa e confusa de interagir, indicando assim que a plataforma testada é mais fácil, interativa e possível de distinguir aonde é que está localizada cada informação necessária.

5.3.1.3. *Experts* do BRaNT

A interação das *experts* do BRaNT com a plataforma foi incremental e contínua ao longo de todo o desenvolvimento do projeto, pois, como referido na secção 3.3, sempre que uma funcionalidade era concluída, a mesma era apresentada as *experts* e a todos os elementos do projeto de investigação. Desta forma recebeu-se *feedback* dos aspetos

funcionais e visuais de cada funcionalidade e conforme o mesmo alterou-se de acordo com as necessidades e prioridades das *experts*.

Um dos exemplos de mudanças foi a incoerência entre os títulos apresentados na plataforma (letras minúsculas em vez de letras maiúsculas e vice-versa) que provocavam inconsistência na mesma. Realizou-se as alterações para que a plataforma estivesse completamente consistente e de acordo com o *feedback* das mesmas.

Todas as funcionalidades consideradas requisitos prioritários foram desenvolvidos e aprovados pelas *experts* e os restantes elementos do projeto. Uma das versões da plataforma que ficou disponível *online*, tal como a API, foram utilizadas no estudo piloto realizado por uma das *experts* do projeto com um paciente real que teve a duração de um mês e meio. Conseguiu-se assim comprovar o sucesso da plataforma e da API desenvolvidas através deste estudo no *mundo real*, sem ocorrência de erros e contratempos.

Realizou-se os testes de usabilidade às *experts* do sistema que indicaram que o que estava em falta era a visualização gráfica da performance nas sessões de treino e no perfil cognitivo. Na tarefa de inserção do programa de treino indicaram que deveriam existir marcadores no *slider* do desafio para a adaptação de 25 em 25.

5.3.2. Avaliação quantitativa

Nesta subsecção será apresentada toda a avaliação quantitativa proveniente dos testes de usabilidade realizados. Serão apresentados todos os resultados do questionário SUS, dividindo-se pelas questões realizadas.

No Gráfico 5.1 estão apresentados os resultados sobre a questão se os participantes gostariam de utilizar a plataforma desenvolvida com frequência.

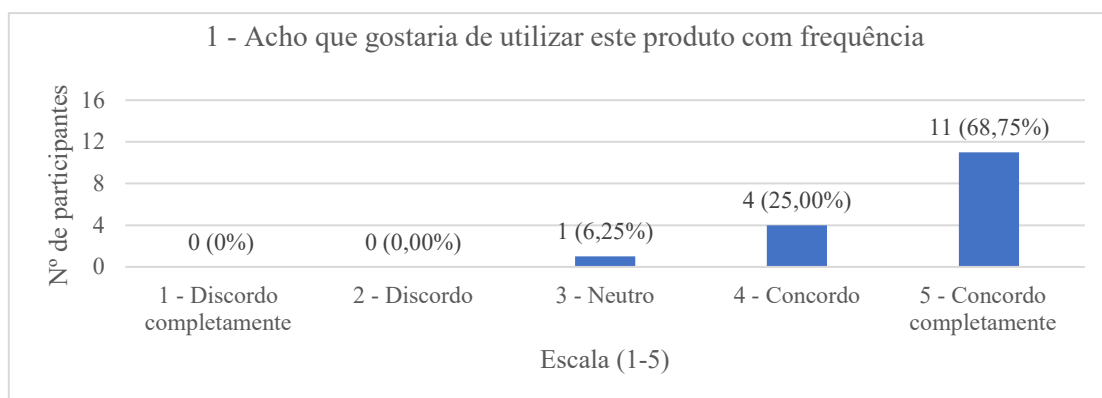


Gráfico 5.1 - Resultados da questão "Acho que gostaria de utilizar este produto com frequência".

Observando o Gráfico 5.1 verifica-se que a maioria dos participantes concordaram que gostariam de utilizar a plataforma com frequência, sendo que apenas um dos participantes manteve-se neutro à sua utilização com frequência. Os resultados positivos nesta questão correspondem ao *feedback* dos participantes, sendo que alguns questionaram se a plataforma não estaria disponível no futuro para utilização.

Os participantes discordaram que a plataforma estava mais complexa do que necessário, sendo que na avaliação qualitativa a simplicidade e o *design* minimalista

foram pontos fortes referidos. É possível observar no Gráfico 5.2 os resultados sobre a questão se os participantes acharam a plataforma mais complexa do que o necessário.

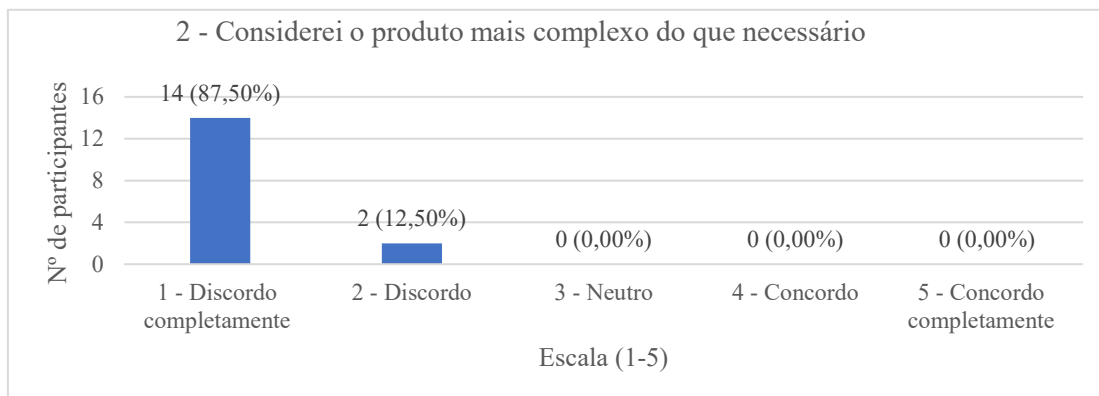


Gráfico 5.2 - Resultados da questão "Considere o produto mais complexo do que necessário".

No Gráfico 5.3 estão os resultados sobre a questão se participantes acharam fácil de utilizar a plataforma testada. Os resultados obtidos indicam que 31,25% dos participantes concordam e 68,75% dos participantes concordam completamente. A partir destas respostas comprova-se as observações gerais realizadas na avaliação qualitativa, onde os participantes referiram que a plataforma era apelativa, intuitiva e perceptível.

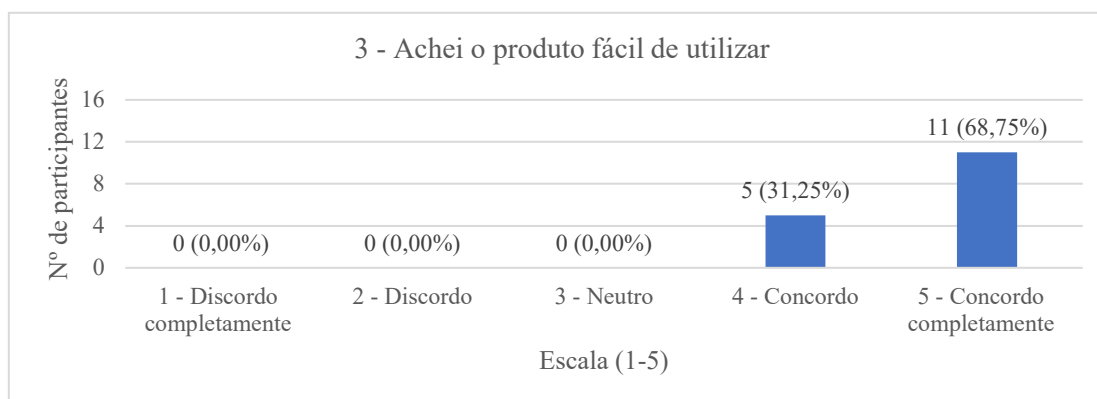


Gráfico 5.3 - Resultados da questão "Achei o produto fácil de utilizar".

Os resultados à questão se os participantes necessitariam de ajuda de um técnico para conseguir utilizar este produto os resultados foram positivos, com 62,5% dos participantes indicarem que discordavam completamente e 25% indicarem que discordavam. Ainda nesta questão um participante foi neutro e um concordou que necessitaria de ajuda para utilizar este produto. Estes resultados vêm do facto que na inserção do programa de treino por vezes alguns participantes, por não ter muita experiência na área da reabilitação cognitiva não tinham certeza que valores inserirem de forma que o programa de treino não fosse algo fora do normal. O mesmo acontece com a inserção dos dados normativos onde o grupo dos estudantes de mestrado, por não terem muita experiência na área, tinham dificuldades em visualizar os dados normativos e de inserir corretamente através da análise realizada. Apesar destas dificuldades não podem ser resolvidas na usabilidade da plataforma, influenciaram nos resultados do gráfico Gráfico 5.4.

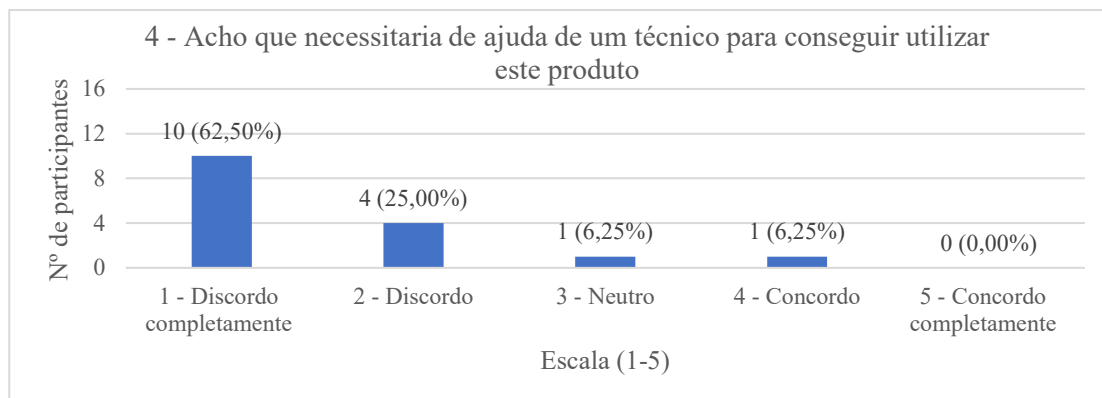


Gráfico 5.4 - Resultados da questão "Acho que necessitaria de ajuda de um técnico para conseguir utilizar este produto".

No Gráfico 5.5 estão apresentados os resultados referentes à pergunta se os participantes consideram que às várias funcionalidades da plataforma estavam bem integradas.

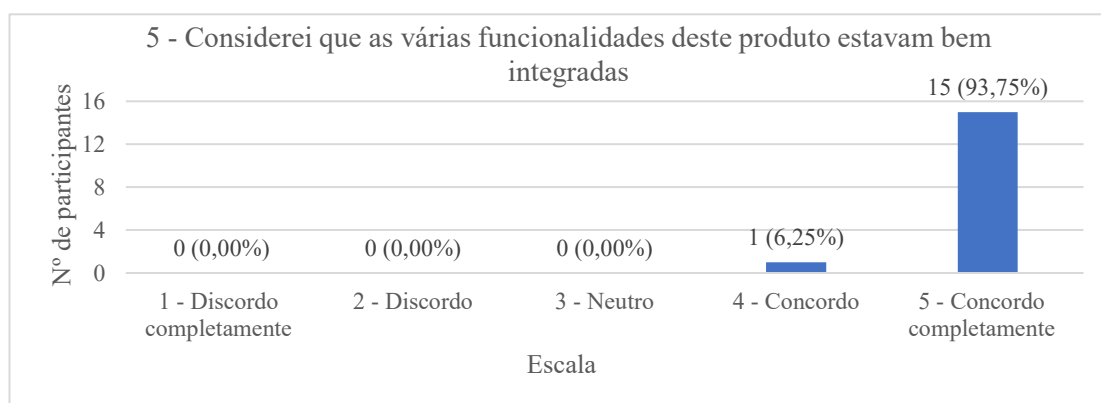


Gráfico 5.5 - Resultados da questão "Considereei que as várias funcionalidades deste produto estavam bem integradas".

Observando o Gráfico 5.5 é possível concluir que as funcionalidades da plataforma estavam bem integradas, pois quinze participantes concordaram completamente e o um participante que concordou. Assim sendo obteve-se uma resposta positiva e correspondente com avaliação quantitativa onde nenhum participante referiu a ocorrência de algum erro que impedisse de realizar as tarefas com sucesso, até quando estavam a realizar a ambientação com a plataforma.

Nos resultados à questão se os participantes acharam que a plataforma tinha muitas inconsistências, 87,5% dos participantes discordaram completamente e 12,5% discordaram que a plataformas continha inconsistências ao longo da sua utilização, sendo que no Gráfico 5.6 estão apresentados estes resultados.

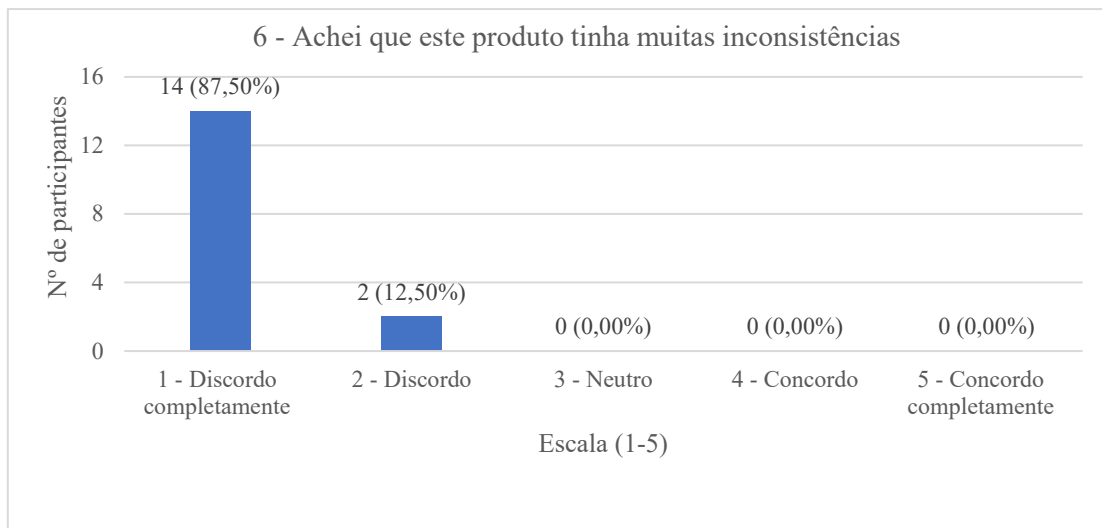


Gráfico 5.6 - Resultados da questão "Achei que este produto tinha muitas inconsistências".

No Gráfico 5.7 estão apresentados os resultados sobre a questão se outros utilizadores aprenderiam utilizar rapidamente a plataforma desenvolvida.

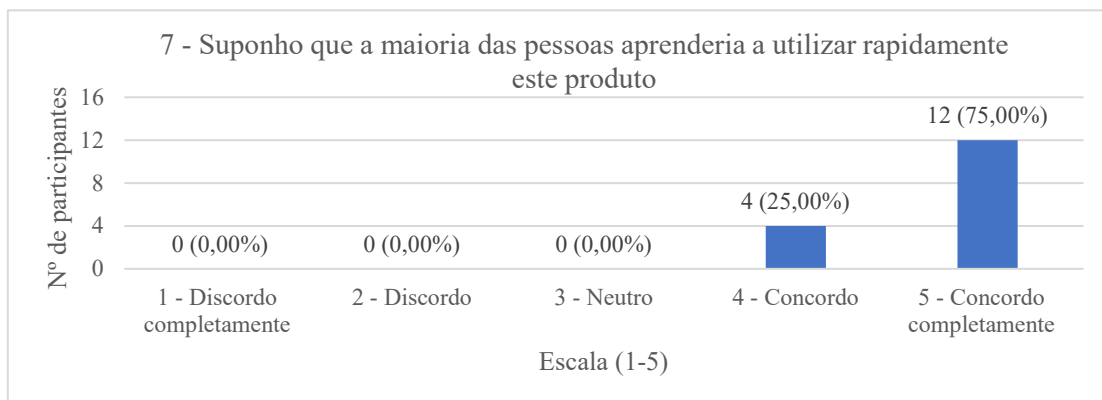


Gráfico 5.7 - Resultados da questão "Suponho que a maioria das pessoas aprenderia a utilizar rapidamente este produto".

Observando o Gráfico 5.7 têm-se que doze participantes concordaram completamente e quatro concordaram que outros utilizadores conseguiram utilizar rapidamente esta plataforma desenvolvida. Tal como na avaliação qualitativa os utilizadores consideraram a plataforma fácil de interação e a utilização do menu e da barra de navegação auxiliou na organização das informações, obtendo um acesso rápido às mesmas.

Os resultados acerca da pergunta se os participantes acharem que a plataforma era muito complicada de utilizar estão apresentados no Gráfico 5.8. Os participantes discordaram que a plataforma testada era complicada de utilizar, afirmando os resultados obtidos na avaliação qualitativa.

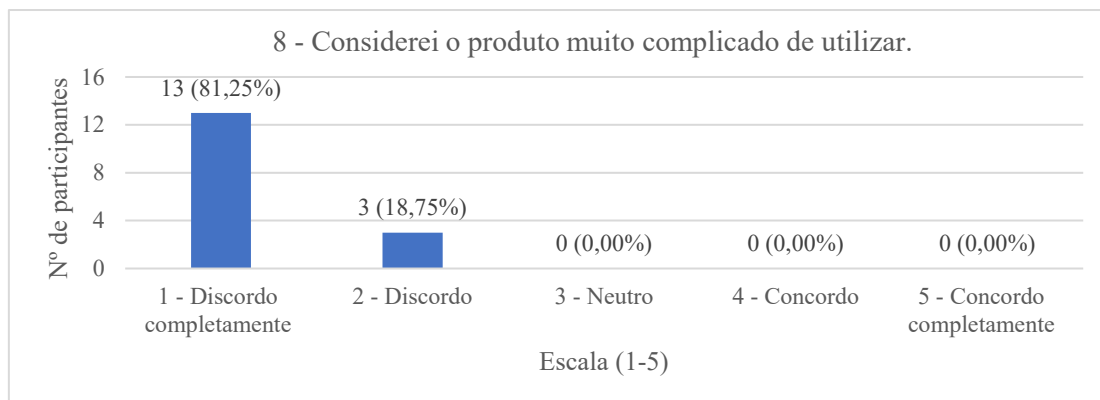


Gráfico 5.8 - Resultados da questão "Considerei o produto muito complicado de utilizar".

No Gráfico 5.9 está apresentado os resultados sobre a questão se os participantes se sentiram muito confiantes ao utilizar a plataforma, sendo que a resposta foi maioritariamente positiva, sentindo-se confiantes na utilização da mesma. Apenas um participante não se sentiu muito confiante a utilizar, sendo que ao analisar os resultados notou-se que o participante era um aluno de mestrado ainda sem experiência na área.

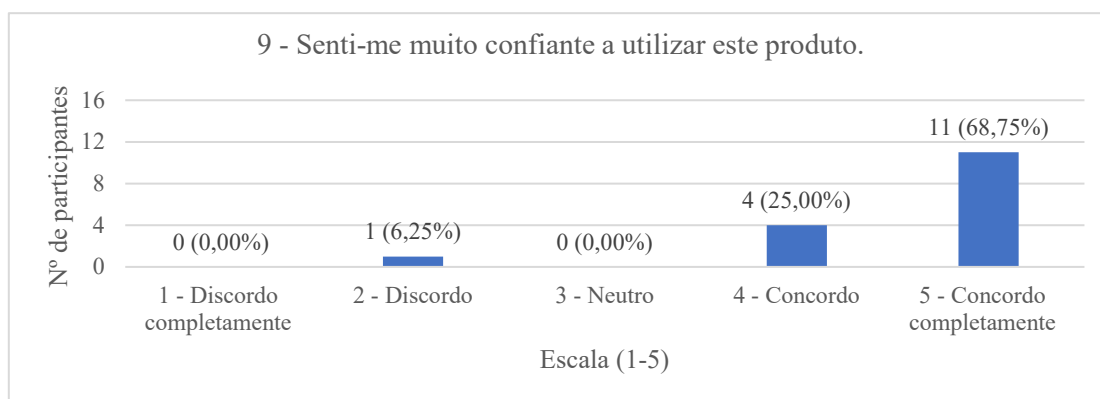


Gráfico 5.9 - Resultados da questão "Senti-me muito confiante a utilizar este produto".

Por fim tem-se os resultados no Gráfico 5.10 da última questão realizada aos participantes, sendo que esta era se estes tiveram de aprender muito antes de lidar com esta plataforma.

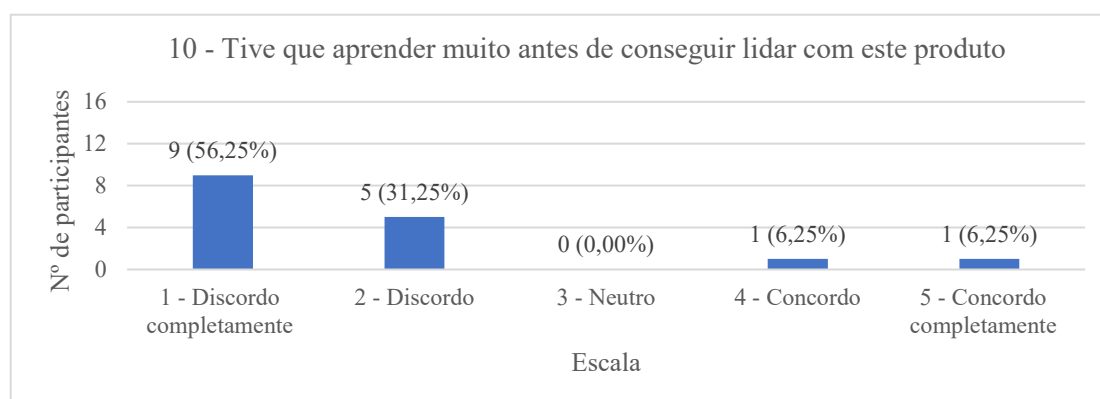


Gráfico 5.10 - Resultados da questão "Tive que aprender muito antes de conseguir lidar com este produto".

Observando o Gráfico 5.10 é possível verificar que mais de metade dos participantes indicaram que não tiveram necessidade de aprender muito antes de conseguir lidar com a plataforma desenvolvida. Apenas dois participantes concordaram que tiveram de aprender muito, isto tendo em conta que os mesmos referiram que esta aprendizagem não seria da plataforma, mas sim habilitações literárias necessárias por serem utilizadores da mesma, por exemplo serem formadas na área da psicologia.

Através das respostas dos participantes a cada questão é possível realizar o cálculo da pontuação de cada participante, que varia entre 0 e 100 pontos, sendo este calculado através da seguinte fórmula [54]:

$$((Q1-1)+(5-Q2)+(Q3-1)+(5-Q4)+(Q5-1)+(5-Q6)+(Q7-1)+(5-Q8)+(Q9-1)+(5-Q10)) \times 2,5$$

No Gráfico 5.11 estão apresentadas as pontuações dos resultados do questionário SUS por participante, calculados através da fórmula acima apresentada.

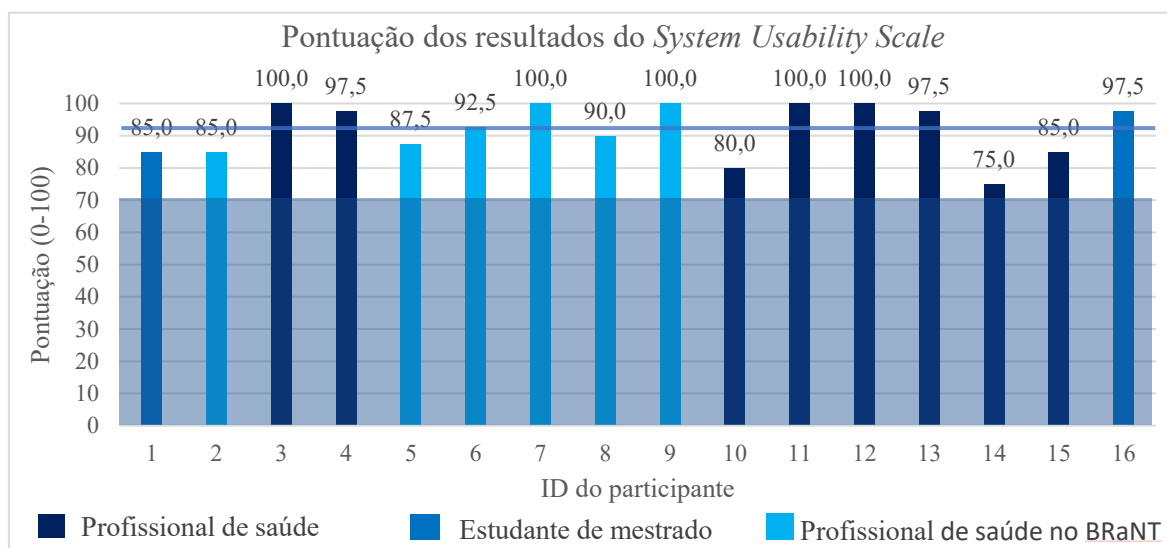


Gráfico 5.11 - Pontuação dos resultados do System Usability Scale.

Observando o Gráfico 5.11 é possível verificar que o valor mais abaixo obtido no questionário foi de 75 pontos e o mais alto 100 pontos, tendo-se assim uma média de 92 pontos. Segundo (ref ref ref) o produto é considerado aceitável quando a sua pontuação varia entre o 70 e 100 pontos. A média do produto encontra-se acima dos 90 pontos sendo este então considerado “Excelente” nas classificações adjetivas, segundo [55], [56].

5.4. Conclusões

De um modo geral, nas avaliações qualitativas e quantitativas, obteve-se resultados positivos e semelhantes em ambas as avaliações, sendo estes resultados favoráveis ao parecer das *experts* e dos elementos do projeto BRaNT. Apesar de atualmente a plataforma desenvolvida não ser um produto que será distribuído foi essencial perceber se existiram aspetos e funcionalidades que deveriam ser modificados ou acrescentados, de modo a ter uma plataforma mais abrangente e completa do ponto de vista funcional e na área da psicologia, tendo em conta a experiência na área que os participantes tinham.

Ao longo desta avaliação, através dos testes de usabilidade, não foram detetados erros funcionais e técnicos na plataforma como na API, indo ao encontro dos testes funcionais e de integração realizados durante o desenvolvimento do projeto.

6. Discussão

Nesta secção será apresentada a discussão acerca da solução proposta, requisitos implementados e contribuições para a revisão de literatura na secção 2.2.1. Na solução proposta foram referidas duas partes: plataforma *web* (NexusBRaNT) e a API, sendo que esta discussão também irá se dividir nessas mesmas partes.

6.1. NexusBRaNT

A plataforma desenvolvida tinha como principal objetivo (secção 2.2.3) facilitar os PS a acederem ao *backoffice* do BRaNT e a gestão dos dados dos pacientes, sendo que esta foi realizada com sucesso. As funcionalidades desenvolvidas foram divididas em quatro grupos principais: gestão de pacientes, gestão de avaliações neuropsicológicas, gestão de programas de treino cognitivo e gestão de utilizadores. Para cada funcionalidade, em cada grupo, desenvolveu-se a inserção, a visualização, a edição e a eliminação de dados, tal como está apresentado na secção 4.4.5.

Os requisitos foram priorizados juntamente com as *experts*. Posteriormente a implementação de uma funcionalidade, esta era apresentada às *experts* e priorizava-se novamente os próximos requisitos a serem implementados. O processo de priorização dos requisitos com as *experts* foi interativo, pois por vezes as necessidades das mesmas modificavam ao longo do desenvolvimento da plataforma. Como decidiu-se proceder ao desenvolvimento deste projeto através de um método ágil, mantendo a comunicação contínua com as *experts* e restantes elementos do projeto, foi possível realizar a priorização dos requisitos de forma interativa, caso contrário não seria possível, e no final da implementação as necessidades das *experts* e elementos do projeto seriam outras.

Começou-se a implementação com gestão do paciente (inserção e visualização da informação sociodemográfica e clínica), gestão de avaliações neuropsicológicas (inserção e visualização das avaliações) e gestão de programas de treino (inserção e visualização de um programa). Posto isto, seguiu-se para a implementação dos menus apresentados na Fig. 4.21. Em seguida, realizou-se as funcionalidades de edição da informação sociodemográfica e clínica, da avaliação neuropsicológica e do programa de treino. A funcionalidade do perfil cognitivo foi implementada paralelamente com a edição dos dados. A visualização das sessões de treino e progresso do programa foram as funcionalidades implementadas em sequência. Após esta implementação foi iniciado o estudo piloto realizado por uma das *experts* do projeto, no qual esta versão da plataforma ficou *online*. Enquanto o estudo piloto decorreu, continuou-se a implementação da plataforma, sendo as funcionalidades de eliminação dos dados, inserção da interpretação clínica e funcionalidades do administrador foram desenvolvidas.

Tendo em conta os requisitos funcionais apresentados na secção 3.1.3 é possível verificar que 85% dos requisitos elicitados foram implementados. Os requisitos que ficaram em falta foi o registo dos instrumentos de avaliação nas funcionalidades do administrador, a representação gráfica de alguns dados e a personalização das atividades cognitivas, deste modo os requisitos RF4, RF11.4, RF12.2, RF15.2 a RF15.4 não foram desenvolvidos. Tendo em conta os resultados obtidos nos testes de usabilidade e a aprovação dos elementos do projeto BRaNT, comprova-se que as funcionalidades

desenvolvidas foram bem implementadas. Os comentários e funcionalidades referidos, pelos participantes, como vantajosos abrangem maior parte dos requisitos implementados. As funcionalidades em falta refletem os requisitos não implementados. Verifica-se também que aplicação contém diversas funcionalidades esperadas pelos PS, que tornam a aplicação muito funcional e as em falta são pormenores que seriam mais um complemento à plataforma. Também é possível verificar que a usabilidade da plataforma está de acordo com as solicitações do grupo de investigação e também indo ao encontro da perceção dos participantes nos testes de usabilidade.

Um dos requisitos do sistema era distinção das funcionalidades comuns acedidas pelos PS clínicos ou investigadores e pelos administradores da plataforma. Deste modo foi necessário criar endereços distintos e restringir quem teria acesso aos mesmos. Caso um PS comum tente aceder ao endereço da plataforma que apenas do administrador tem acesso, é apresentado no ecrã uma mensagem que a página pretendida não foi encontrada, o mesmo se aplica a um administrador tentar aceder às funcionalidades de um PS comum. Na lateral esquerda do ecrã apresenta-se o menu para que o PS comum ou o administrador possa aceder às funcionalidades disponibilizadas em cada caso.

Tal como foi referido na secção 4.4.4, foram utilizados componentes reutilizáveis para implementação da plataforma. A utilização destes componentes, também dinâmicos, facilitaram quando novos requisitos eram elicitados e refinados. Por exemplo, após a implementação do formulário de inserção de uma avaliação neuropsicológica as *experts* do projeto solicitaram a inserção de dois novos instrumentos de avaliação. Tendo em conta que a plataforma é escalável, com campos de inserção dinâmicos, foi apenas necessária a inserção dos dados dos instrumentos na base de dados e a criação de duas tabelas na mesma para armazenar os dados inseridos desses instrumentos e, por consequência a criação das rotas, as funções nos controladores e os modelos, no lado da API. Outro exemplo dos componentes dinâmicos é nos campos de seleção, onde os dados apresentados nos mesmos provêm da base de dados. Quando é necessário ter mais uma opção no campo de seleção é só necessário adicionar o mesmo na tabela correspondente da base de dados. Deste modo é só necessário a ver um único componente de seleção que recebe os dados que deverão ser apresentados.

Outro ponto referido na secção 1.3, acerca do NexusBRaNT, é que este deveria ser possível de ser acedido em qualquer lugar (casa, trabalho ou espaço público com acesso à *internet*) e em qualquer dispositivo (computador, *tablet* ou telemóvel). Assim sendo desenvolveu-se a plataforma de forma que seja responsiva e funcional em todos os dispositivos e colocou-se *online*.

6.2. API

A segunda parte da solução proposta foi a implementação de uma API que dá suporte ao NexusBRaNT e a sistemas externos do projeto BRaNT. O maior objetivo desta API é dar resposta aos pedidos provenientes da plataforma desenvolvida e às atividades cognitivas desenvolvidas noutro sistema.

Tal como foi referido na secção 4.3, desenvolveu-se as funções para corresponder às solicitações de inserção, atualização, eliminação e obtenção de dados. Seguiu-se a

arquitetura MVC, onde os componentes modelos e controladores estão presentes no lado da API e o componente *view* do lado da plataforma.

Em relação às rotas utilizadas na plataforma desenvolvida, as mesmas eram implementadas paralelamente ao *frontend*, conforme as necessidades de inserção, de atualização, de eliminação e de obtenção de dados. Os testes realizados às rotas criadas foram realizados na plataforma e na ferramenta PostMan (referida na secção 4.1).

Para o desenvolvimento das rotas que respondem a outro sistema do projeto BRaNT, necessitou-se de reunir com o elemento do projeto responsável pelo desenvolvimento das atividades cognitivas no Unity. Começou-se por analisar quais as necessidades e solicitações que seriam realizadas para as atividades cognitivas. Posto isto, definiu-se as rotas do tipo *get* e implementou-se as mesmas. Os testes realizados às rotas para o sistema externo foram realizados na ferramenta PostMan e também com o elemento responsável que testava as rotas no sistema implementado. Após as rotas do tipo *get*, definiu-se as necessidades de inserção e atualização juntamente com o elemento responsável. Posteriormente a estas definições, implementou-se as rotas solicitadas e testou-se as mesmas através do processo referido acima. Foi necessário definir, com o elemento responsável, o formato que seria enviado o objeto json para API para a inserção desses dados.

Além dos testes de usabilidade acima referidos, o sistema desenvolvido foi utilizado num estudo piloto (no âmbito do projeto de investigação) por uma *expert*, durante um mês e meio. Este estudo piloto foi realizado a um paciente, onde quais foram realizadas avaliações neuropsicológicas e sessões de treino, sendo esta informação inserida e visualizada no NexusBRaNT. Após o mês e meio de testes e de utilização da plataforma e, por consequência, da API, é possível concluir o sucesso da plataforma e da API desenvolvidas, tanto no lado da plataforma como no lado do sistema externo (atividades cognitivas).

6.3. Contribuições para a área

Será também interessante comparar as oportunidades e os desafios dos HIS (apresentadas na secção 1) com a plataforma desenvolvida. O NexusBRaNT também processa e transmite os dados remotamente. Deste modo, o PS pode aceder em qualquer parte aos dados dos seus pacientes, melhorando a qualidade de atendimento dos mesmos. O NexusBRaNT também poderá reduzir os erros, pois existem campos que somam a totalidade dos campos anteriores automaticamente, apresentado esse resultado aos PS. O cálculo do perfil cognitivo e da interpretação clínica facilitam também os PS, pois, para o cálculo dos mesmos, são necessárias fórmulas mais complexas para sua realização. Assim sendo, o sistema calculando automaticamente o perfil cognitivo e a interpretação clínica evitam erros humanos. Alguns desafios apresentados dos HIS foram as diferentes necessidades dos PS e a resistência na utilização destes sistemas. A plataforma desenvolvida, ao ser focada apenas numa área da saúde, sendo o público-alvo os psicólogos, neuropsicólogos, entre outros que trabalham na área da reabilitação cognitiva, foi possível abranger e aprofundar as diversas necessidades que os PS têm. Tendo em conta que os sistemas de gestão de pacientes e reabilitação cognitiva apresentados (secção 2.2.1) focam-se noutros utilizadores, as funcionalidades desenvolvidas não são totalmente

concentradas para gestão e monitorização dos pacientes, pois também contém os jogos cognitivos. Atualmente a utilização de tecnologias na área da saúde é maior e, tendo em conta a situação pandémica que ocorreu, a resistência dos PS em utilizarem tecnologias foi reduzida. Através da avaliação qualitativa e quantitativa concluiu-se que a utilização desta plataforma seria vantajosa para os participantes, tornando o seu trabalho mais eficiente e de certa forma mais concentrado, tendo toda a informação dos seus pacientes numa única plataforma.

Tendo em conta os sistemas específicos analisados na secção 2.2.1 e a Tabela 2.1, pode-se comparar os sistemas referidos com o sistema desenvolvido. O público-alvo no NexusBRaNT são apenas os PS, enquanto nos outros sistemas são os PS, pacientes e outros utilizadores. A plataforma desenvolvida contém funcionalidades comuns aos outros sistemas: gestão de paciente e gestão de programa de treinos. Tendo em conta que o sistema apenas se concentra nos PS existe um grupo de funcionalidades que nos outros sistemas não existe, como por exemplo a gestão das avaliações neuropsicológicas e o perfil cognitivo, sendo estas uma vantagem para sistema existente, focando-se nas necessidades dos PS aumentado a eficiência dos mesmos, facilitando a gestão e monitorização remota dos seus pacientes.

7. Conclusão

Neste trabalho foi desenvolvida uma plataforma *web* e uma API que dá suporte a um sistema de reabilitação cognitiva, auxiliando os PS a gerirem e monitorizarem informações, avaliações neuropsicológicas e programas treino dos seus pacientes.

Foi importante todo o processo anterior à implementação da plataforma e da API, sendo estes: análise de requisitos antigos, elicitación, refinamento e priorização dos novos requisitos. Os casos de utilização facilitaram a compreensão de todo o sistema e a interação entre os tipos de utilizadores e a plataforma. Optou-se pelo método de desenvolvimento interativo e incremental, tendo havido a possibilidade de comunicação contínua com as *experts* do projeto e os restantes elementos. Desta forma, as alterações realizadas na plataforma não foram tão complexas, também não seria possível pois a priorização de requisitos e elicitación de novos foram surgindo ao longo do desenvolvimento.

O desenvolvimento deste projeto envolveu a base de dados, a API e o NexusBRaNT. Começou-se por corrigir a base de dados, realizando as alterações necessárias referidas na secção 4.2 para ir de encontro com as informações elicitadas nos requisitos. As implementações da API e da plataforma foram realizadas paralelamente. Primeiro desenvolvia-se o *frontend* de uma funcionalidade e posteriormente, consoante as necessidades, desenvolvia-se as rotas e as funções necessárias. Realizou-se desta forma pois era necessário decidir como seria enviado o objeto *json*, no caso de inserção, para API e assim inserir esses mesmos dados na base de dados. Também foi mais fácil entender o que cada funcionalidade precisava e só em seguida realizar as solicitações para a API. Após terminar a implementação de uma funcionalidade ou função (definição da rota e função do controlador/modelo) da API, testava-se com diversos dados as mesmas, a fim de detetar erros. Caso a funcionalidade na plataforma passasse nos testes, a mesma era apresentada a todos os elementos do BRaNT (reuniões semanais) ou nos *sprints* com as *experts* para aprovação da mesma.

Para além do sucesso da plataforma e API no estudo piloto realizado por uma das *experts* do projeto, os resultados dos testes de usabilidade com os 16 participantes foram muito positivos. Apesar dos diferentes grupos entre os participantes, as observações acerca da plataforma foram semelhantes, tendo como vantagem as seguintes funcionalidades: a variedade instrumentos de avaliação à escolha, os dados normativos e interpretação clínica nas avaliações neuropsicológicas, o cálculo automático do perfil cognitivo e da interpretação clínica e a visualização da performance e progresso nas sessões de treinos. De modo geral os participantes referiram que a plataforma estava intuitiva, fácil de utilização, perceptível e apelativa. A organização das informações apresentadas na plataforma foi também um ponto forte referido pelos participantes. As funcionalidades extras referidas pelos participantes e os requisitos que faltaram desenvolver serão apresentados em seguida como trabalho futuro.

Uma das dificuldades sentidas na realização deste trabalho foi a seleção de informação a ser apresentada no mesmo, sendo que ao longo do trajeto académico são poucos os trabalhos no quais necessitamos de investigação prévia antes do desenvolvimento de um projeto. O facto de a base de dados também estar parcialmente

desenvolvida foi necessário realizar uma análise da mesma antes de continuar com as suas alterações, sendo que foram esclarecidas diversas dúvidas acerca das tabelas e suas relações. A gestão de tempo e tarefas também foi um desafio, pois as tarefas também tinham um prazo interno do projeto de investigação para estarem concluídas. O facto de conciliar trabalhar em *frontend*, *backend* e dar suporte a outro sistema externo à plataforma existente, foi necessária uma boa gestão de tarefas e tempo, sendo estas dificuldades foram superadas.

Uma das oportunidades que este trabalho oferece é o desenvolvimento de um projeto com diversos conceitos e técnicas aprendidos ao longo da trajetória académica. Outra oportunidade no desenvolvimento desta tese é o trabalho em equipa com membros de outras áreas, possibilitando adaptação na comunicação e partilha de conhecimentos. O facto de o projeto de mestrado envolver diversas tarefas, desenvolveu-se competências em diversas áreas (gestão do projeto, implementação em *backend* e em *frontend*), que noutro projeto poderia não ser tão abrangente.

7.1. Trabalho futuro

A próxima etapa do projeto será a priorização dos requisitos em falta e a eliciação das novas funcionalidades referidas pelos participantes dos testes de usabilidade, caso sejam essenciais no âmbito do projeto de investigação. Tendo em conta os requisitos em falta e às funcionalidades indicadas pelos participantes, o trabalho futuro é o seguinte:

- O administrador conseguir registar novos instrumentos de avaliação neuropsicológica na plataforma (nome do instrumento, informação detalhada dos campos e interpretação clínica caso exista);
- Visualização gráfica da performance absoluta, da interpretação gráfica e da performance das sessões de treino;
- Visualização de um resumo das atividades cognitivas realizadas pelos pacientes, indicando quais os domínios predominantes de cada atividade;
- Reagendar a sessão de treino ou interromper um programa de treino, envolvendo também o sistema externo que foi desenvolvido;
- Visualização das recomendações do sistema de reabilitação, alteração parâmetros de intervenção e seleção das atividades cognitivas desejadas que o paciente necessita de realizar;
- Registo de informações sobre o familiar ou cuidador que será o contacto de emergência do paciente.

8. Referências

- [1] R. S. Ledley e L. B. Lusted, «Reasoning Foundations of Medical Diagnosis», *Sci. New Ser.*, vol. 130, n.º 3366, pp. 9–21, 1959.
- [2] E. H. Shortliffe e J. J. Cimino, Eds., *Biomedical informatics: computer applications in health care and biomedicine*, 3rd ed. New York, NY: Springer, 2006.
- [3] R. M. Gardner, T. A. Pryor, e H. R. Warner, «The HELP hospital information system: update 1998», *Int. J. Med. Inf.*, vol. 54, n.º 3, pp. 169–182, jun. 1999, doi: 10.1016/S1386-5056(99)00013-1.
- [4] Gilad J. Kuperman, Reed M. Gardner, e T. Allan Pryor, *HELP: A Dynamic Hospital Information System*. Springer New York. Acedido: 16 de outubro de 2022. [Em linha]. Disponível em: <https://link.springer.com/book/10.1007/978-1-4612-3070-0>
- [5] G. O. Barnett *et al.*, «COSTAR—A computer-based medical information system for ambulatory care», *Proc. IEEE*, vol. 67, n.º 9, pp. 1226–1237, set. 1979, doi: 10.1109/PROC.1979.11438.
- [6] E. Vaganova, T. Ishchuk, A. Zemtsov, e D. Zhdanov, «Health Information Systems: Background and Trends of Development Worldwide and in Russia:», em *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies*, Porto, Portugal, 2017, pp. 424–428. doi: 10.5220/0006244504240428.
- [7] J. H. Holmes, R. Bellazzi, L. Sacchi, e N. Peek, Eds., *Artificial Intelligence in Medicine: 15th Conference on Artificial Intelligence in Medicine, AIME 2015, Pavia, Italy, June 17-20, 2015. Proceedings*, vol. 9105. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-19551-3.
- [8] R. Haux, «Health information systems – past, present, future», *Int. J. Med. Inf.*, vol. 75, n.º 3, pp. 268–281, mar. 2006, doi: 10.1016/j.ijmedinf.2005.08.002.
- [9] R. C. Barrows e P. D. Clayton, «Privacy, confidentiality, and electronic medical records.», *J. Am. Med. Inform. Assoc.*, vol. 3, n.º 2, pp. 139–148, 1996.
- [10] P. G. Goldschmidt, «HIT and MIS: implications of health information technology and medical information systems», *Commun. ACM*, vol. 48, n.º 10, pp. 68–74, out. 2005, doi: 10.1145/1089107.1089141.
- [11] A. S. Aula, «HOSPITAL MANAGEMENT INFORMATION SYSTEMS ACCEPTANCE AT WONOSARI REGIONAL HOSPITAL, GUNUNGKIDUL, YOGYAKARTA, INDONESIA», *Epidemiol. Soc. Health Rev. ESHR*, vol. 3, n.º 1, p. 15, mar. 2021, doi: 10.26555/eshr.v3i1.3594.
- [12] M. Khosrow-Pour, D.B.A., Ed., *Encyclopedia of Information Science and Technology, Third Edition*: IGI Global, 2015. doi: 10.4018/978-1-4666-5888-2.
- [13] P. Pagalday-Olivares, B. A. Sjöqvist, J. Adjordor-van de Beek, S. Abudey, A. R. Silberberg, e R. Buendia, «Exploring the feasibility of eHealth solutions to decrease delays in maternal healthcare in remote communities of Ghana», *BMC Med. Inform. Decis. Mak.*, vol. 17, n.º 1, p. 156, dez. 2017, doi: 10.1186/s12911-017-0552-z.
- [14] L. M. M. Gonçalves, «Plataforma web para gestão e visualização de dados de um sistema de reabilitação cognitiva», masterThesis, 2021. Acedido: 9 de novembro de 2021. [Em linha]. Disponível em: <https://digituma.uma.pt/handle/10400.13/3285>
- [15] S. Kasper *et al.*, «Management of mild cognitive impairment (MCI): The need for national and international guidelines», *World J. Biol. Psychiatry*, vol. 21, n.º 8, pp. 579–594, set. 2020, doi: 10.1080/15622975.2019.1696473.
- [16] S. Gauthier *et al.*, «Mild cognitive impairment», *Lancet Lond. Engl.*, vol. 367, n.º 9518, pp. 1262–1270, abr. 2006, doi: 10.1016/S0140-6736(06)68542-5.

- [17] S. Ataollahi Eshkoo, C. Y. Mun, C. K. Ng, e T. A. Hamid, «Mild cognitive impairment and its management in older people», *Clin. Interv. Aging*, p. 687, abr. 2015, doi: 10.2147/CIA.S73922.
- [18] J. M. Torpy, C. Lynn, e R. M. Glass, «Mild Cognitive Impairment», *JAMA*, vol. 302, n.º 4, p. 452, jul. 2009, doi: 10.1001/jama.302.4.452.
- [19] S. Jongsiriyanong e P. Limpawattana, «Mild Cognitive Impairment in Clinical Practice: A Review Article», *Am. J. Alzheimers Dis. Dementiasr*, vol. 33, n.º 8, pp. 500–507, dez. 2018, doi: 10.1177/1533317518791401.
- [20] «Cogweb - Treino cognitivo online». <https://www.cogweb.pt/> (acedido 3 de dezembro de 2021).
- [21] «CogniFit», *Teste Neuropsicológico e Estimulação Cognitiva I Exercícios mentais, treino cerebral profissional*. <https://www.cognifit.com/pt> (acedido 3 de dezembro de 2021).
- [22] «NeuronUP. Web platform of cognitive rehabilitation ★★★★★». <https://neuronup.us/> (acedido 6 de dezembro de 2021).
- [23] «RehaCom - Terapia cognitiva | HASOMED GmbH». <https://hasomed.de/es/productos/rehacom-terapia-cognitiva/> (acedido 18 de outubro de 2022).
- [24] «Solution digitale de référence en stimulation, rééducation et remédiation cognitives pour les professionnels de santé», *HappyNeuron Pro*. <https://www.happyneuronpro.com/?noredirect> (acedido 10 de dezembro de 2021).
- [25] V. T. Cruz *et al.*, «A Rehabilitation Tool Designed for Intensive Web-Based Cognitive Training: Description and Usability Study», *JMIR Res. Protoc.*, vol. 2, n.º 2, p. e59, dez. 2013, doi: 10.2196/resprot.2899.
- [26] V. Tedim Cruz *et al.*, «Web-Based Cognitive Training: Patient Adherence and Intensity of Treatment in an Outpatient Memory Clinic», *J. Med. Internet Res.*, vol. 16, n.º 5, p. e122, mai. 2014, doi: 10.2196/jmir.3377.
- [27] L. B. Gouveia e J. Ranito, «SISTEMAS DE INFORMAÇÃO DE APOIO À GESTÃO», p. 96.
- [28] «Classter - All-in-one School & Learning Management Platform», *Classter - All-in-one School & Learning Management System*. <https://www.classter.com/> (acedido 9 de dezembro de 2021).
- [29] SPOTMARKET, «PAAE – Software de Gestão Escolar», *PAAE – Software de Gestão Escolar*. <https://paae.pt/> (acedido 9 de dezembro de 2021).
- [30] «Software de Recursos Humanos all-in-one - FactorialHR». <https://factorialhr.pt/> (acedido 10 de dezembro de 2021).
- [31] «uMan Xpert | Software Recursos Humanos | Gestão do Capital Humano», *uMan Xpert HR Solutions*. <https://uman.pt/> (acedido 10 de dezembro de 2021).
- [32] «Laravel - The PHP Framework For Web Artisans». <https://laravel.com/> (acedido 21 de dezembro de 2021).
- [33] «CakePHP - Build fast, grow solid | PHP Framework | Home», *CakePHP - The rapid development php framework*. <https://cakephp.org/> (acedido 27 de dezembro de 2021).
- [34] «The web framework for perfectionists with deadlines | Django». <https://www.djangoproject.com/> (acedido 21 de dezembro de 2021).
- [35] «Spring makes Java simple.», *Spring*. <https://spring.io/> (acedido 21 de dezembro de 2021).
- [36] «Express - Node.js web application framework». <https://expressjs.com/> (acedido 21 de dezembro de 2021).
- [37] «Vue.js». <https://vuejs.org/> (acedido 27 de dezembro de 2021).

- [38] «Angular». <https://angular.io/> (acedido 27 de dezembro de 2021).
- [39] «Ember.js - A framework for ambitious web developers». <https://emberjs.com/> (acedido 27 de dezembro de 2021).
- [40] J. F.- js.foundation, «jQuery». <https://jquery.com/> (acedido 30 de dezembro de 2021).
- [41] «React – A JavaScript library for building user interfaces». <https://reactjs.org/> (acedido 27 de dezembro de 2021).
- [42] J. Niederst Robbins, *Learning Web design: a beginner's guide to HTML, CSS, JavaScript, and web graphics*, Fourth edition. Beijing: O'Reilly, 2012.
- [43] S. McCool, *Laravel starter: the definitive introduction to the Laravel PHP web development framework*. Birmingham, UK: Packt Pub, 2013.
- [44] D. Rubio, *Beginning Django*. Berkeley, CA: Apress, 2017. doi: 10.1007/978-1-4842-2787-9.
- [45] R. Johnson *et al.*, «Spring Framework Reference Documentation», p. 847.
- [46] A. Mardan, *Express.js Guide: The Comprehensive Book on Express.js*. Azat Mardan, 2014.
- [47] A. Silberschatz, H. F. Korth, e S. Sudarshan, *Database system concepts*, vol. 4. McGraw-Hill New York, 1997.
- [48] N. Jatana, S. Puri, M. Ahuja, I. Kathuria, e D. Gosain, «A Survey and Comparison of Relational and Non-Relational Database», *Int. J. Eng. Res.*, vol. 1, n.º 6, p. 5, 2012.
- [49] A. Nayak, «Type of NOSQL Databases and its Comparison with Relational Databases», *Int. J. Appl. Inf. Syst.*, vol. 5, p. 4, 2013.
- [50] «MoCA - Cognitive Assessment», *MoCA – Cognitive Assessment*. <https://www.mocatest.org/> (acedido 12 de janeiro de 2022).
- [51] K. E. Wiegers e J. Beatty, *Software requirements*, Third edition. Redmond, Washington: Microsoft Press, s division of Microsoft Corporation, 2013.
- [52] L. Bass, P. Clements, e R. Kazman, *Software architecture in practice*, 3rd ed. Upper Saddle River, NJ: Addison-Wesley, 2013.
- [53] W. L. in R.-B. U. Experience, «Thinking Aloud: The #1 Usability Tool», *Nielsen Norman Group*. <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/> (acedido 7 de novembro de 2022).
- [54] J. Brooke, «SUS - A quick and dirty usability scale».
- [55] A. Bangor, «Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale», vol. 4, n.º 3, 2009.
- [56] A. Bangor, P. T. Kortum, e J. T. Miller, «An Empirical Evaluation of the System Usability Scale», *Int. J. Hum.-Comput. Interact.*, vol. 24, n.º 6, pp. 574–594, jul. 2008, doi: 10.1080/10447310802205776.

Anexos

Anexo A – Informações base de dados

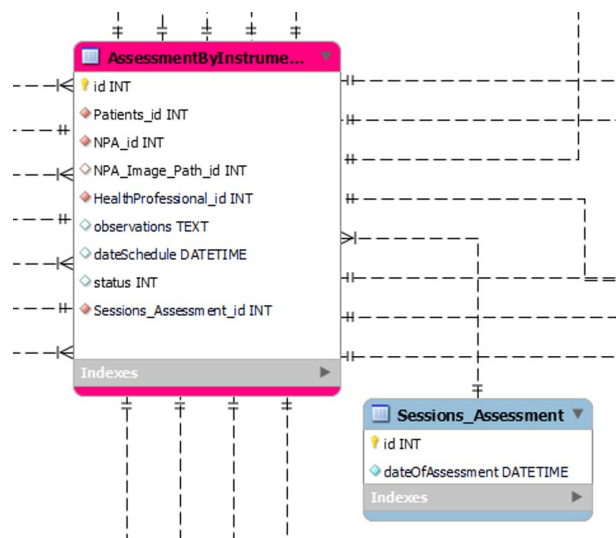


Fig. A.1 – Relação de um para muitos nas tabelas "Sessions_Assessment" e "AssessmentByInstrument".

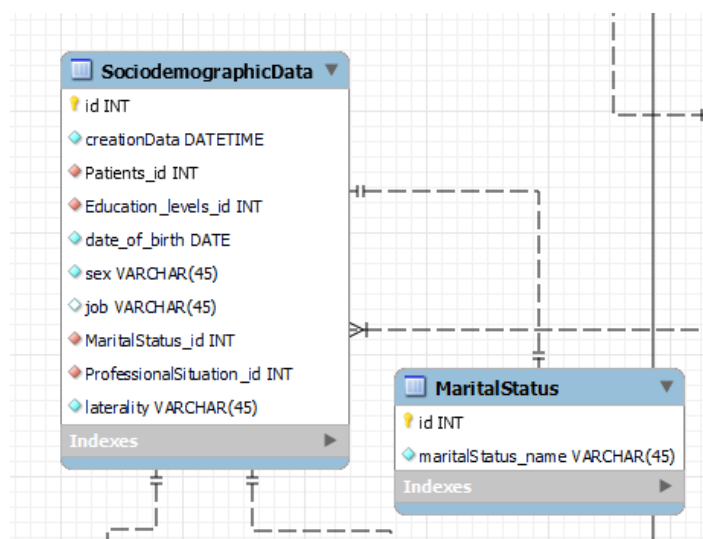


Fig. A.2 – Tabela "Marital Status" adicionada à BD e ligação de 1 para 1 com a tabela "SociodemographicData".

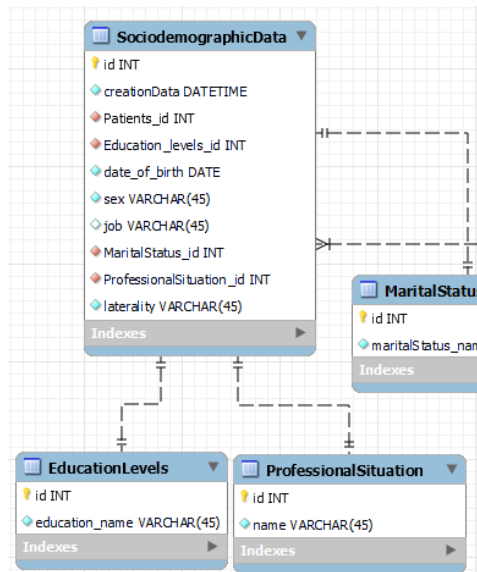


Fig. A.3 – Tabela "ProfessionalSituation" adicionada à base de dados e ligação de 1 para 1 com a tabela "SociodemographicData".

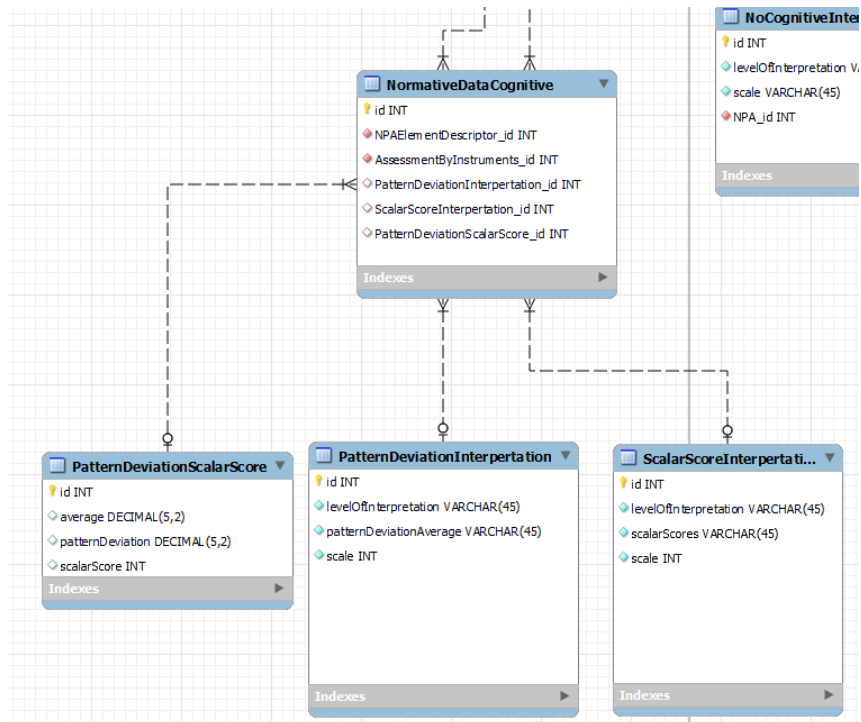


Fig. A.4 – Tabela "NormativeDataCognitive" e tabelas associadas "PatternDeviationScalarScore", "PatternDeviationInterperation" e "ScalarScoreInterperati...".

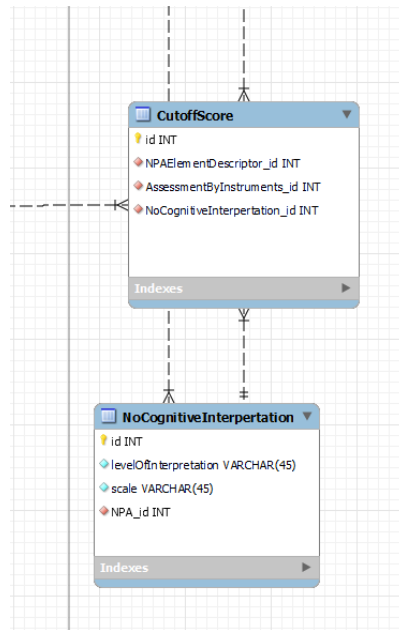


Fig. A.5 – Tabela "CutoffScore" e tabela associada "NoCognitiveInterperation".

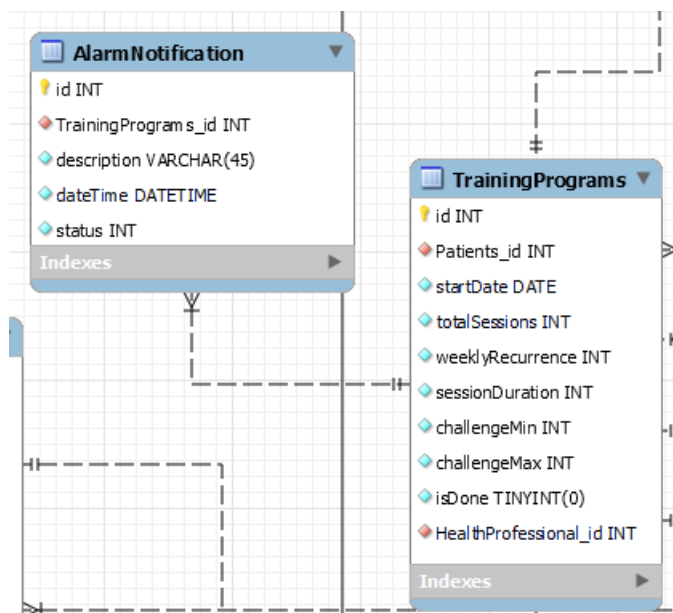


Fig. A.6 – Tabela "AlarmNotification" e sua relação com a tabela "TrainingPrograms".

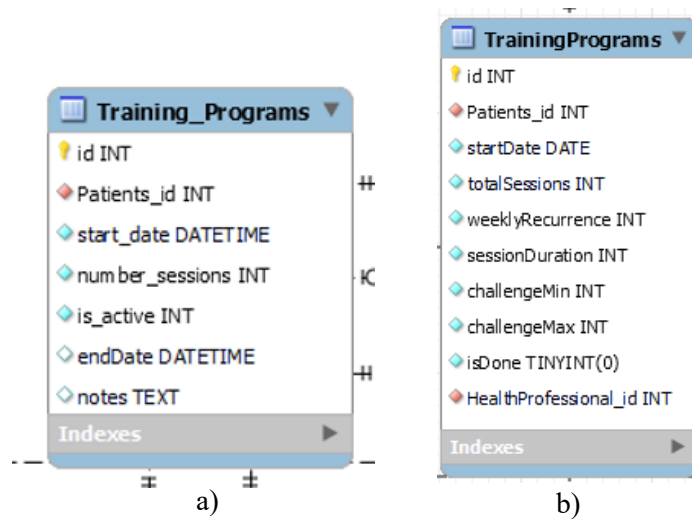


Fig. A.7 – Tabela “*Training_Programs*”: a) antiga; b) atual.

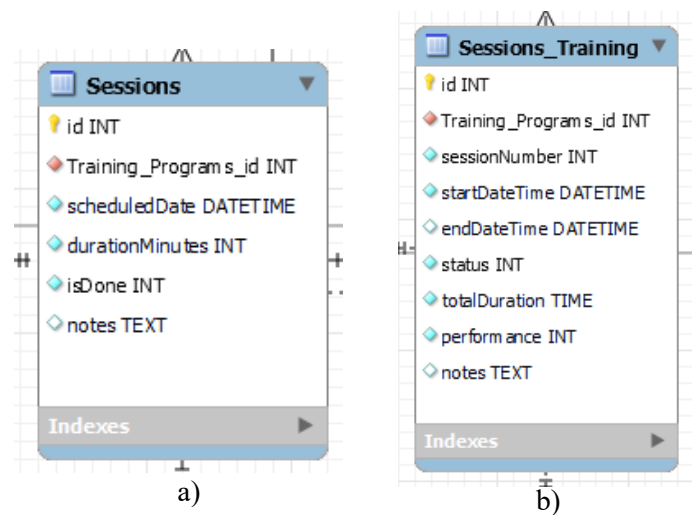


Fig. A.8 – Tabela “*Sessions_Training*”: a) antiga; b) atual.

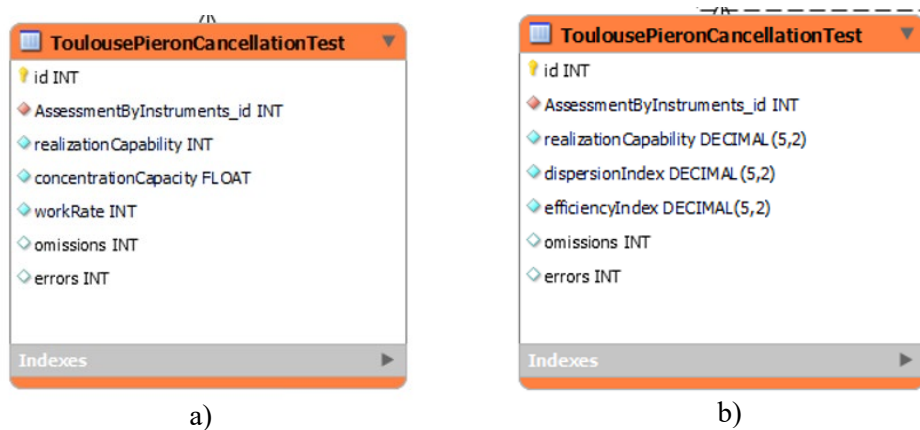


Fig. A.9 – Tabela “*ToulousePieronCancellationTest*”: a) antiga; b) atual.

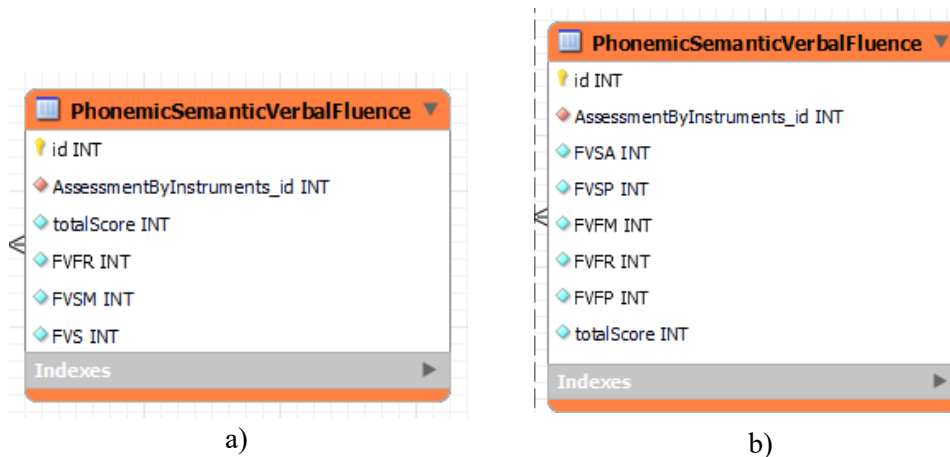


Fig. A.10 – Tabela "PhonemicSemanticVerbalFluence": a) antiga; b) atual.

Tabela A.1 - Instrumento de avaliação BDI-II.

BDI-II			
Nome do campo	Campo na BD	Min.	Máx.
BDI-II	BDIII	0	63

Tabela A.2 - Instrumento de avaliação CDR.

CDR			
Nome do campo	Campo na BD	Min.	Máx.
Memória	memory	0	3
Orientação	orientation	0	3
Juízo e resolução de problemas	judgmentProblemSolving	0	3
Atividades nas comunidades	communityActivities	0	3
Casa e passatempos	homeHobbies	0	3
Cuidado pessoal	personalCare	0	3
Total	totalScore	0	18
Observações do NPA			
Total calculado automaticamente (somadas de todos os campos anteriores)			
1 casa decimal nos campos			

Tabela A.3 - Instrumento de avaliação FCSRT.

FCSRT			
Nome do campo	Campo na BD	Min.	Máx.
free immediate recall - Evocação livre imediata	FIR	0	48
free cued recall - Evocação guiada imediata	FCR	0	48
total immediate recal - Total evocação imediata (livre + guiada)	TIR	0	48
free delayed recal - Evocação livre diferida	FDR	0	16
free cued delayed recall - Evocação guiada diferida	FCDR	0	16
total delayed recall - Total evocação diferida (livre + guiada)	TDR	0	48

Tabela A.4 - Instrumento de avaliação Figura Complexa de Rey.

Figura Complexa de Rey			
Nome do campo	Campo na BD	Min.	Máx.
Cópia	copyResult	0	36
Memória imediata	immediateMemoryResult	0	36
Memória diferida	defferedMemoryResult	0	36

Tabela A.5 - Instrumento de avaliação Fluências Verbais Semânticas e Fonéticas.

Fluências Verbais Semânticas e Fonéticas			
Nome do campo	Campo na BD	Min.	Máx.
Fluências verbais semânticas - Animais	FVSA	-	-
Fluências verbais semânticas - Profissões	FVSP	-	-
Fluências verbais fonética M	FVFM	-	-
Fluências verbais fonética R	FVFR	-	-
Fluências verbais fonética P	FVFP	-	-
Fluências verbais fonética - Total	totalScore	-	-
Observações do NPA			
Total calculado automaticamente (somadas dos campos FVFM + FVFR + FVFP)			

Tabela A.6 - Instrumento de avaliação GATSB.

GATSB			
Nome do campo	Campo na BD	Min.	Máx.
GATSB	scoreTotal	0	58

Tabela A.7 - Instrumento de avaliação GDS-15.

GDS-15			
Nome do campo	Campo na BD	Min.	Máx.
GDS-15	score15	0	15

Tabela A.8 - Instrumento de avaliação GDS-30.

GDS-30			
Nome do campo	Campo na BD	Min.	Máx.
GDS-30	score30	0	30

Tabela A.9 - Instrumento de avaliação PRECiS.

PRECiS			
Nome do campo	Campo na BD	Min.	Máx.
PRECiS	scoreTotal	0	128

Tabela A.10 - Instrumento de avaliação IAFAI.

IAFAI			
Nome do campo	Campo na BD	Min.	Máx.
ABVD	ABVD	0	100
AIVD-F	AIVD-F	0	100
AIVD-A	AIVD-A	0	100
Total	totalScore	0	100
Observações do NPA			
Total calculado automaticamente (somadas de todos os campos anteriores)			
2 casas decimais nos campos			

Tabela A.11 - Instrumento de avaliação MOCA.

MOCA			
Nome do campo	Campo na BD	Min.	Máx.
Visuo-espacial / executiva	visoSpatialExecutive	0	5
Nomeação	naming	0	3
Atenção	attentionTotal	0	6
Linguagem	language	0	3
Abstração	abstraction	0	2
Evocação diferida	delayedRecall	0	5
Orientação	orientation	0	6
Total	totalScore	0	30
Observações do NPA			
Total calculado automaticamente (somadas de todos os campos anteriores)			

Tabela A.12 - Instrumento de avaliação QOLIBRI.

QOLIBRI			
Nome do campo	Campo na BD	Min.	Máx.
Total	total	0	100
Cognitivo	cognitive	0	100
Self	self	0	100
AVD's	avds	0	100
Relações sociais	socialRelationships	0	100
Emoções	emotions	0	100
Condição física	physicalCondition	0	100
Observações do NPA			
2 casas decimais nos campos			

Tabela A.13 - Instrumento de avaliação QSM.

QSM			
Nome do campo	Campo na BD	Min.	Máx.
QSM	scoreTotal	0	21

Tabela A.14 - Instrumento de avaliação Rey15.

Rey15			
Nome do campo	Campo na BD	Min.	Máx.
Ensaio da evocação imediata	immediateEvocationScore	0	15
Resultado combinado do reconhecimento	combinedReconitionScore	0	30

Tabela A.15 - Instrumento de avaliação SIS.

SIS			
Nome do campo	Campo na BD	Min.	Máx.
Força	strength	0	100
Força da mão	handStrength	0	100
Mobilidade	mobility	0	100
Atividades da vida diária	dailyLiving	0	100
Emocional	emotional	0	100
Memória	memory	0	100
Comunicação	communication	0	100
Participação social	socialParticipation	0	100
Recuperação do AVC	strokeRecoveryRate	0	100
Observações do NPA			
2 casas decimais nos campos			

Tabela A.16 - Instrumento de avaliação Toulouse-Piéron.

Toulouse-Piéron			
Nome do campo	Campo na BD	Min.	Máx.
Capacidade de realização	realizationCapability	0	37.5
Índice de dispersão	dispersionIndex	0	100
Índice de eficiência	efficiencyIndex	0	100
Observações do NPA			
2 casas decimais nos campos			

Tabela A.17 - Instrumento de avaliação WAIS-II.

WAIS-III			
Nome do campo	Campo na BD	Min.	Máx.
Pesquisa de símbolos	symbolSearch	0	60
Código	code	0	133
Vocabulário	vocabulary	0	66

Tabela A.18 - Instrumento de avaliação WHOQOL-BREF.

WHOQOL-BREF			
Nome do campo	Campo na BD	Min	Máx
Domínio 1	domain1	0	100
Domínio 2	domain2	0	100
Domínio 3	domain3	0	100
Domínio 4	domain4	0	100

Tabela A.19 - Instrumento de avaliação WHOQOL-OLD.

WHOQOL-OLD			
Nome do campo	Campo na BD	Min.	Máx.
Funcionamento sensorial	sensorialFunction	0	20
Autonomia	autonomy	0	20
Actividade passadas / presentes / futuras	activitysPPF	0	20
Participação social	socialParticipation	0	20
Morte / morrer	death	0	20
Intimidade	intimacy	0	20
Família e vida familiar	familyLife	0	20
Total	totalScore	0	140
Observações do NPA			
Total calculado automaticamente (somadas de todos os campos anteriores)			

Tabela A.20 - Instrumento de avaliação WMS.

WMS			
Nome do campo	Campo na BD	Min	Máx
Recordação imediata	immediateRecall	0	104
Recordação longo prazo	longTermRecall	0	104
Cópia	copy	0	104
Discriminação	discrimination	0	7
Reconhecimento	recognition	0	48
Percentagem de retenção	yieldPercentage	0	100

Anexo A.1 – Base de dados antiga

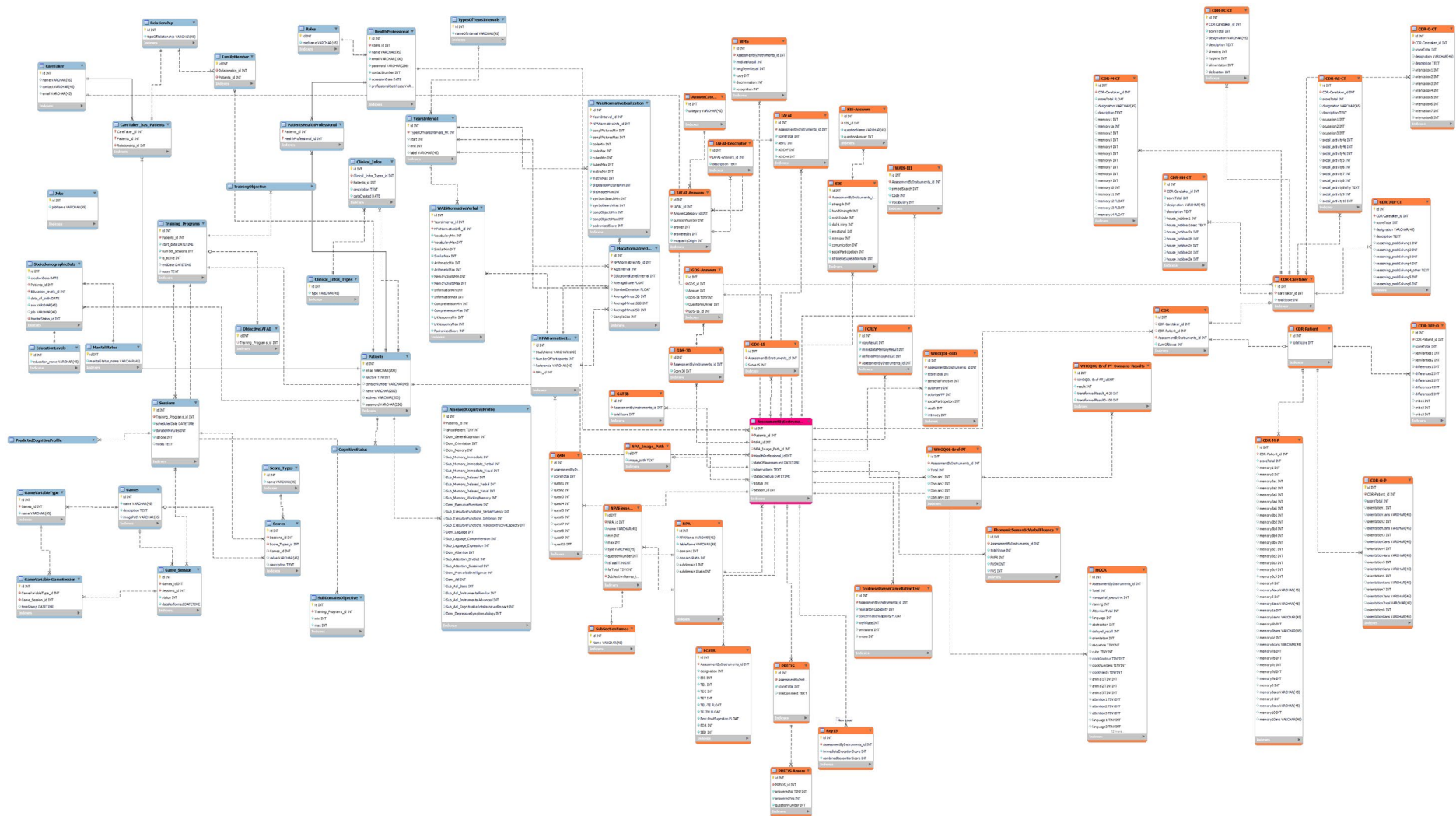


Fig. A.11 – Base de dados antiga do projeto de investigação BRaNT.

Tabela A.21 - Tabela "Patients" na base de dados.

Tabela "Patients": Armazena informações sobre o paciente						
Atributos						
Nome	Tipo	Não nulo	PK	FK	Default	Significado
id	INT	Sim	Sim	Não		
email	VARCHAR(200)	Sim	Não	Não		
isActive	TINYINT	Sim	Não	Não	1	0 - false 1 - true
contactNumber	VARCHAR(45)	Não	Não	Não		
name	VARCHAR(200)	Não	Não	Não		
address	VARCHAR(200)	Não	Não	Não		
password	VARCHAR(256)	Não	Não	Não		

Tabela A.22 - Tabela "NPA" na base de dados.

Tabela "NPA": Tabela intermédia entre "AssessmentByInstrument" e "NPASelementDescriptor". Contém os nomes dos NPA e em nome da tabela que contém os resultados para cada NPA. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
NPAName	VARCHAR(45)	Sim	Não	Não	Nome do NPA aplicado
tableName	VARCHAR(45)	Sim	Não	Não	O nome deste campo é o nome exato da tabela de instrumento destino
domain1	INT	Não	Não	Não	
domain1Ratio	INT	Não	Não	Não	
subdomain1	INT	Não	Não	Não	
subdomain1Ratio	INT	Não	Não	Não	

Tabela A.23 - Tabela "HealthProfessional" na base de dados.

Tabela "HealthProfessional": Armazena informações sobre o profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Roles id	INT	Sim	Não	Sim	
name	VARCHAR(45)	Sim	Não	Não	
email	VARCHAR(100)	Sim	Não	Não	
password	VARCHAR(256)	Sim	Não	Não	
contactNumber	INT	Sim	Não	Não	
accessionDate	DATE	Sim	Não	Não	Data de registo do PS
professionalCertificate	VARCHAR(100)	Não	Não	Não	

Tabela A.24 - Tabela "SociodemographicData" na base de dados.

Tabela "SociodemographicData": Dados sociodemográficos dos pacientes. Possuem uma data associada que representa a data de atualização. A função de atualização não deve ser usada nesta tabela, uma nova entrada com uma nova data deve ser feita, para que o histórico seja mantido					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
creationData	DATETIME	Sim	Não	Não	
Patients_id	INT	Sim	Não	Sim	
Education_levels_id	INT	Sim	Não	Sim	
date of birth	DATE	Sim	Não	Não	
sex	VARCHAR(45)	Sim	Não	Não	
job	VARCHAR(45)	Sim	Não	Não	
MaritalStatus_id	INT	Sim	Não	Sim	
ProfessionalSituation_id	INT	Sim	Não	Sim	
laterality	VARCHAR(45)	Sim	Não	Não	

Tabela A.25 - Tabela "AssessmentByInstruments" na base de dados.

Tabela "AssessmentByInstruments": Cruza as informações de um resultado de NPA, um paciente e um profissional de saúde que realizou o exame.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Patients_id	INT	Sim	Não	Sim	
NPA_id	INT	Sim	Não	Sim	ID do NPA que foi realizado
NPA_Image_Path_id	INT	Não	Não	Sim	Endereço da imagem do NPA inserida
HealthProfessional_id	INT	Sim	Não	Sim	
observations	TEXT	Não	Não	Não	Campo de texto aberto para o profissional de saúde colocar notas relevantes em relação à avaliação do NPA
dateSchedule	DATETIME	Não	Não	Não	
status	INT	Não	Não	Não	0 - faltou; 1 - presente; 2 - cancelou.
Sessions_Assessment_id	INT	Sim	Não	Sim	

Tabela A.26 - Tabela "MOCA" na base de dados.

Tabela "MOCA" - Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde. As entradas não obrigatórias, serão trabalhos futuros, podem aumentar grandemente o desempenho da AI.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
visoSpatialExecutive	INT	Sim	Não	Não	Campo Visuo-Espacial/Executiva Min: 0 Max: 5
naming	INT	Sim	Não	Não	Campo Nomeação Min: 0 Max: 3
attentionTotal	INT	Sim	Não	Não	Campo Atenção Min: 0 Max: 6
language	INT	Sim	Não	Não	Campo Linguagem Min: 0 Max: 3
abstraction	INT	Sim	Não	Não	Campo Abstração Min: 0 Max: 2
delayedRecall	INT	Sim	Não	Não	Campo Evocação Diferida Min: 0 Max: 5
orientation	INT	Sim	Não	Não	Campo Orientação Min: 0 Max: 6
totalScore	INT	Sim	Não	Não	Campo Pontuação Total Min: 0 Max: 30
sequence	TINYINT	Não	Não	Não	
cube	TINYINT	Não	Não	Não	
clockContour	TINYINT	Não	Não	Não	
clockNumbers	TINYINT	Não	Não	Não	
clockHands	TINYINT	Não	Não	Não	
animal1	TINYINT	Não	Não	Não	
animal2	TINYINT	Não	Não	Não	
animal3	TINYINT	Não	Não	Não	
attention1	TINYINT	Não	Não	Não	
attention2	TINYINT	Não	Não	Não	
attention3	TINYINT	Não	Não	Não	
language1	TINYINT	Não	Não	Não	
language2	TINYINT	Não	Não	Não	
orientation1	TINYINT	Não	Não	Não	
orientation2	TINYINT	Não	Não	Não	
orientation3	TINYINT	Não	Não	Não	
orientation4	TINYINT	Não	Não	Não	
orientation5	TINYINT	Não	Não	Não	
orientation6	TINYINT	Não	Não	Não	
verbalFluency	INT	Não	Não	Não	
abstraction1	INT	Não	Não	Não	
abstraction2	INT	Não	Não	Não	
delaydEvocation1	INT	Não	Não	Não	
delaydEvocation1Category	TINYINT	Não	Não	Não	
DelaydEvocation1MultipleChoice	TINYINT	Não	Não	Não	

Tabela A.27 - Tabela "IAFAI" na base de dados.

Tabela "IAFAI": Armazena os resultados do IAFAI. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments id	INT	Sim	Não	Sim	
ABVD	DECIMAL(5,2)	Sim	Não	Não	Campo ABVD Min: 0 Max: 100%
AIVD-F	DECIMAL(5,2)	Sim	Não	Não	Campo AIVD-F Min: 0 Max: 100%
AIVD-A	DECIMAL(5,2)	Sim	Não	Não	Campo AIVD-A Min: 0 Max: 100%
totalScore	DECIMAL(5,2)	Sim	Não	Não	Campo Pontuação Total Min: 0 Max: 100%

Tabela A.28 - Tabela "CDR" na base de dados.

Tabela "CDR": Tabela utilizada para cruzar todas as informações entre as respostas dadas pelos pacientes. Resultados obrigatórios preenchidos pelo profissional de saúde. As entradas não obrigatórias, serão trabalhos futuros, podem aumentar grandemente o desempenho da AI.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments id	INT	Sim	Não	Sim	
memory	DECIMAL(5,1)	Sim	Não	Não	Campo Memória Min: 0 Máx: 3
orientation	DECIMAL(5,1)	Sim	Não	Não	Campo Orientação Min: 0 Máx: 3
judgmentProblemSolving	DECIMAL(5,1)	Sim	Não	Não	Campo Juízo e Resolução de Problemas Min: 0 Máx: 3
communityActivities	DECIMAL(5,1)	Sim	Não	Não	Campo Atividades na Comunidades Min: 0 Máx: 3
homeHobbies	DECIMAL(5,1)	Sim	Não	Não	Campo Casa e Passatempos Min: 0 Máx: 3
personalCare	DECIMAL(5,1)	Sim	Não	Não	Campo Cuidado Pessoal Min: 0 Máx: 3
totalScore	DECIMAL(5,1)	Sim	Não	Não	Pontuação máxima Min: 0 Máx: 18
CDR-Caretaker id	INT	Não	Não	Sim	
CDR-Patient id	INT	Não	Não	Sim	

Tabela A.29 - Tabela "GATSB" na base de dados.

Tabela "GATSB": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
scoreTotal	INT	Sim	Não	Não	Campo GATSB Min: 0 Max: 100

Tabela A.30 - Tabela "GDS_30" na base de dados.

Tabela "GDS_30": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
score30	INT	Sim	Não	Não	Campo GDS30 Min: 0 Max: 30

Tabela A.31 - Tabela "WAIS_III" na base de dados.

Tabela "WAIS_III": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
symbolSearch	INT	Sim	Não	Não	Campo Pesquisa de Símbolos Min: 0 Max: 60
code	INT	Sim	Não	Não	Campo Código Min: 0 Max: 133
vocabulary	INT	Sim	Não	Não	Campo Vocabulário Min: 0 Max: 66

Tabela A.32 - Tabela "QSM" na base de dados.

Tabela "QSM": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde. As entradas não obrigatórias, serão trabalhos futuros, podem aumentar grandemente o desempenho da AI.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
scoreTotal	INT	Sim	Não	Não	Campo Pontuação Total Min: 0 Max: 21
quest1	INT	Não	Não	Não	0 – Não; 1 - Sim, mas não importante; 2 - Sim, com alguma importância; 3 - Sim, com problemas.
quest2	INT	Não	Não	Não	0 – Não; 1 - Por vezes; 2 – Frequentemente.
quest3	INT	Não	Não	Não	0 - Não; 1 - Sim, mas não importante; 2 - Sim, com alguma importância; 3 - Sim, com problemas.
quest4	INT	Não	Não	Não	0 - Não; 1 - Sim, mas não importante; 2 - Sim, com alguma importância; 3 - Sim, com problemas.
quest5	INT	Não	Não	Não	0 – Não; 1 - Por vezes; 2 – Frequentemente.
quest6	INT	Não	Não	Não	0 – Não; 1 – Sim.
quest7	INT	Não	Não	Não	0 – Não; 1 – Sim.
quest8	INT	Não	Não	Não	0 - Não; 1 - Sim; 2 - Sim, com problemas.
quest9	INT	Não	Não	Não	0 - Não; 1 - Sim; 2 - Sim, com problemas.
quest10	INT	Não	Não	Não	0 - Não; 1 - Sim; 2 - Sim, com problemas.

Tabela A.33 - Tabela "EducationLevels" na base de dados.

Tabela "EducationLevels": Armazena informação sobre os níveis de educação. Esta informação é introduzida manualmente.						
Atributos						
Nome	Tipo	Não nulo	PK	FK	Default	Significado
id	INT	Sim	Sim	Não		
education_name	VARCHAR(45)	Sim	Não	Não		Ex.: 0, 1, 2 ...

Tabela A.34 - Tabela "WHOQOL_OLD" na base de dados.

Tabela "WHOQOL_OLD": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde. As entradas não obrigatórias, serão trabalhos futuros, podem aumentar grandemente o desempenho da AI.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
sensorialFunction	INT	Sim	Não	Não	Campo Funcionamento Sensorial Min: 0 Max: 20
autonomy	INT	Sim	Não	Não	Campo Autonomia Min: 0 Max: 20
activitysPPF	INT	Sim	Não	Não	Campo Atividades passadas / presentes / futuras Min: 0 Max: 20
socialParticipation	INT	Sim	Não	Não	Campo Participação Social Min: 0 Max: 20
death	INT	Sim	Não	Não	Campo Morte Min: 0 Max: 20
intimacy	INT	Sim	Não	Não	Campo Intimidade Min: 0 Max: 20
familyLife	INT	Sim	Não	Não	Campo Família e Vida Familiar Min: 0 Max: 20
totalScore	INT	Sim	Não	Não	Campo Pontuação Total Min: 0 Max: 140

Tabela A.35 - Tabela "Rey15" na base de dados.

Tabela "Rey15": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
immediateEvocationScore	INT	Sim	Não	Não	Campo Ensaio da evocação imediata Min: 0 Max: 15
combinedReconitionScore	INT	Sim	Não	Não	Campo Resultado combinado do reconhecimento Min: 0 Max: 30

Tabela A.36 - Tabela "WMS" na base de dados.

Tabela "WMS": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
immediateRecall	INT	Sim	Não	Não	Campo Recordação imediata Min: 0 Max: 104
longTermRecall	INT	Sim	Não	Não	Campo Recordação logo prazo Min: 0 Max: 104
copy	INT	Sim	Não	Não	Campo Cópia Min: 0 Max: 104
discrimination	INT	Sim	Não	Não	Campo Discriminação Min: 0 Max: 7
recognition	INT	Sim	Não	Não	Campo Reconhecimento Min: 0 Max: 48
yieldPercentage	INT	Sim	Não	Não	

Tabela A.37 - Tabela "ToulousePieronCancellationTest" na base de dados.

Tabela "ToulousePieronCancellationTest": - Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
realizationCapability	DECIMAL(5,2)	Sim	Não	Não	Campo Capacidade de realização Min: 0 Max: 37.5 %
dispersionIndex	DECIMAL(5,2)	Sim	Não	Não	Campo Índice de Dispersão Min: 0 Max: 100 %
efficiencyIndex	DECIMAL(5,2)	Sim	Não	Não	Campo Índice de Eficiência Min: 0 Max: 100 %
omissions	INT	Não	Não	Não	Not answered (miss)
errors	INT	Não	Não	Não	Wrongly crossed

Tabela A.38 - Tabela "FCSRT" na base de dados.

Tabela "FCSRT": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
FIR	INT	Sim	Não	Não	Campo Evocação livre imediata Min: 0 Máx: 16
FCR	INT	Sim	Não	Não	Campo Evocação guiada imediata Min: 0 Máx: 16
TIR	INT	Sim	Não	Não	Campo Total evocação imediata (livre + guiada) Min: 0 Máx: 48
FDR	INT	Sim	Não	Não	Campo Evocação livre diferida Min: 0 Máx: 16
FCDR	INT	Sim	Não	Não	Campo Evocação guiada diferida Min: 0 Máx: 16
TDR	INT	Sim	Não	Não	Campo Total evocação diferida (livre + guiada) Min: 0 Máx: 16

Tabela A.39 - Tabela "FCREY" na base de dados.

Tabela "FCREY": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
copyResult	DECIMAL(5,1)	Sim	Não	Não	Campo Cópia Min: 0 Máx: 36
immediateMemoryResult	DECIMAL(5,1)	Sim	Não	Não	Campo Memória Imediata Min: 0 Máx: 36
defferedMemoryResult	DECIMAL(5,1)	Sim	Não	Não	Campo Memória Diferida Min: 0 Máx: 36

Tabela A.40 - Tabela "*PhonemicSemanticVerbalFluence*" na base de dados.

Tabela " <i>PhonemicSemanticVerbalFluence</i> ": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
FVSA	INT	Sim	Não	Não	Fluências Verbais Semânticas - Animais Sem max. e sem min.
FVSP	INT	Sim	Não	Não	Fluências Verbais Semânticas - Profissões Sem max.e sem min.
FVFM	INT	Sim	Não	Não	Fluências Verbais Fonética M Sem max e sem min
FVFR	INT	Sim	Não	Não	Fluências Verbais Fonética R Sem max e sem min
FVFP	INT	Sim	Não	Não	Fluências Verbais Fonética P Sem max e sem min
totalScore	INT	Sim	Não	Não	

Tabela A.41 - Tabela "*WHOQOL_Bref_PT*" na base de dados.

Tabela " <i>WHOQOL_Bref_PT</i> ": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
domain1	INT	Sim	Não	Não	Campo Domínio 1 Min: 0 Max: 100
domain2	INT	Sim	Não	Não	Campo Domínio 2 Min: 0 Max: 100
domain3	INT	Sim	Não	Não	Campo Domínio 3 Min: 0 Max: 100
domain4	INT	Sim	Não	Não	Campo Domínio 4 Min: 0 Max: 100

Tabela A.42 - Tabela "*SubSectionNames*" na base de dados.

Tabela " <i>SubSectionNames</i> ": Armazena a informação sobre todos os nomes das subsecções dos nomes nos campos dos NPA's. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Name	VARCHAR(45)	Sim	Sim	Não	

Tabela A. 43 - Tabela "SIS" na base de dados.

Tabela "SIS": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments id	INT	Sim	Não	Sim	
strength	DECIMAL(5,2)	Sim	Não	Não	Campo Força Min: 0 Max: 100 %
handStrength	DECIMAL(5,2)	Sim	Não	Não	Campo Força da Mão Min: 0 Max: 100 %
mobility	DECIMAL(5,2)	Sim	Não	Não	Campo Mobilidade Min: 0 Max: 100 %
dailyLiving	DECIMAL(5,2)	Sim	Não	Não	Campo Atividades da vida diária Min: 0 Max: 100 %
emotional	DECIMAL(5,2)	Sim	Não	Não	Campo Emocional Min: 0 Max: 100 %
memory	DECIMAL(5,2)	Sim	Não	Não	Campo Memória Min: 0 Max: 100 %
comunication	DECIMAL(5,2)	Sim	Não	Não	Campo Comunicação Min: 0 Max: 100 %
socialParticipation	DECIMAL(5,2)	Sim	Não	Não	Campo Participação Social Min: 0 Max: 100 %
strokeRecuperationRate	DECIMAL(5,2)	Sim	Não	Não	Campo Recuperação de AVC Min: 0 Max: 100 %

Tabela A.44 - Tabela "PRECiS" na base de dados.

Tabela "PRECiS": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
scoreTotal	INT	Sim	Não	Não	Campo Pontuação Total Min: 0 Max: 128
finalComment	TEXT	Não	Não	Não	Comentário final que o paciente queria acrescentar.

Tabela A.45 - Tabela "Relationship" na base de dados.

Tabela "Relationship": Armazena informação sobre os tipos de membros da família. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
typeOfRelationship	VARCHAR(45)	Sim	Não	Não	Ex.: Esposa, pai, filho, cuidador informal, etc.

Tabela A.46 - Tabela "Clinical_Infos" na base de dados.

Tabela "Clinical_Infos": Armazena informação sobre a informação clínica dos pacientes.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Clinical_Infos_Types_id	INT	Sim	Não	Sim	
Patients_id	INT	Sim	Não	Sim	
description	TEXT	Sim	Não	Não	
dateCreated	DATE	Sim	Não	Não	

Tabela A.47 - Tabela "Clinical_Infos_Types" na base de dados.

Tabela "Clinical_Infos_Types": Armazena informação sobre os tipos de informação clínica. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
type	VARCHAR(45)	Sim	Não	Não	O nome que identifica este tipo de dados clínicos

Tabela A.48 - Tabela "Roles" na base de dados.

Tabela "Roles": Armazena informação sobre os tipos de utilizadores que existem na plataforma. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
roleName	VARCHAR(45)	Sim	Não	Não	

Tabela A.49 - Tabela "TrainingPrograms" na base de dados.

Tabela "TrainingPrograms": Armazena informação de programas de treino cognitivo.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Patients_id	INT	Sim	Não	Sim	
startDate	DATE	Sim	Não	Não	
totalSessions	INT	Sim	Não	Não	
weeklyRecurrence	INT	Sim	Não	Não	
sessionDuration	INT	Sim	Não	Não	
challengeMin	INT	Sim	Não	Não	Desafio para adaptação – Min: 0
challengeMax	INT	Sim	Não	Não	Max: 60 e Max > Min
isDone	TINYINT(0)	Sim	Não	Não	0 – Em curso; 1- Finalizado.
HealthProfessional_id	INT	Sim	Não	Sim	

Tabela A.50 - Tabela "Sessions_Training" na base de dados.

Tabela "Sessions Training": Armazena informação de sessões de treino cognitivo.						
Atributos						
Nome	Tipo	Não nulo	PK	FK	Default	Significado
id	INT	Sim	Sim	Não		
Training_Programs_id	INT	Sim	Não	Sim		
sessionNumber	INT	Sim	Não	Não		
startDateTime	DATETIME	Sim	Não	Não		Data na qual a sessão deveria ser executada
endDateTime	DATETIME	Não	Não	Não		
status	INT	Sim	Não	Não	0	0 – Aberta; 1 – Parcial; 2 – Completa.
totalDuration	TIME	Sim	Não	Não		
performance	INT	Sim	Não	Não		
notes	TEXT	Não	Não	Não		

Tabela A.51 - Tabela "Activities" na base de dados.

Tabela "Activities": Armazena a informação de cada atividade cognitiva disponível. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
name	VARCHAR(45)	Sim	Não	Não	
description	TEXT	Sim	Não	Não	
imagePath	VARCHAR(45)	Não	Não	Não	Endereço da imagem no servidor

Tabela A.52 - Tabela "ActivitiesVariables" na base de dados

Tabela "ActivitiesVariables": Armazena as variáveis existentes disponíveis em cada atividade, Esta informação será fornecida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
variableName	VARCHAR(45)	Sim	Não	Não	
type	VARCHAR(45)	Sim	Não	Não	

Tabela A.53 - Tabela "ActivitiesIterations" na base de dados.

Tabela "ActivitiesIterations": Armazena a informação das atividades atribuídas às sessões.						
Atributos						
Nome	Tipo	Não nulo	PK	FK	Default	Significado
id	INT	Sim	Sim	Não		
Sessions_id	INT	Sim	Não	Sim		
Activities_id	INT	Sim	Não	Sim		
dateTimeConcluded	DATETIME	Não	Não	Não		
performance	INT	Não	Não	Não	NULL	
contentId	INT	Não	Não	Não		

Tabela A.54 - Tabela "ActivitiesEvents" na base de dados.

Tabela "ActivitiesEvents": Armazena a informação das variáveis durante a execução de uma iteração de atividade.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
ActivitiesVariables_id	INT	Sim	Não	Sim	
ActivitiesIterations_id	INT	Sim	Não	Sim	
value	VARCHAR(50)	Sim	Não	Não	
dateTime	DATETIME	Sim	Não	Não	

Tabela A.55 - Tabela "FamilyMember" na base de dados.

Tabela "FamilyMember": Armazena informação sobre as relações entre familiares e pacientes.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Relationship_id	INT	Sim	Não	Sim	
Patients_id	INT	Sim	Não	Sim	
creationData	DATETIME	Sim	Não	Não	

Tabela A.56 - Tabela "NPAAElementDescriptor" na base de dados.

Tabela "NPAAElementDescriptor": Contém todos os elementos de todos os NPA's. Descreve os campos de cada NPA's: nome, valor mínimo, valor máximo e tipo. Isto é utilizado como tabela de apoio para construir as formas dos NPA's. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
NPA id	INT	Sim	Não	Sim	
name	VARCHAR(45)	Sim	Não	Não	Nome do elemento no NPA
min	FLOAT(5,2)	Sim	Não	Não	Valor mínimo para este elemento do NPA
max	FLOAT(5,2)	Sim	Não	Não	Valor máximo para este elemento do teste
type	VARCHAR(45)	Sim	Não	Não	Descreve o tipo no caso de não ser do tipo numérico nulo
questionNumber	INT	Sim	Não	Não	Determina a ordem no elemento no formulário
forTotal	TINYINT	Não	Não	Não	O campo conta para a soma dos totais: 0 - false (não conta); 1 - true (conta); NULL – não tem campo do total.
isNull	TINYINT	Sim	Não	Não	O campo é nulo ou não: 0 - false (não é nulo); 1 - true (é nulo).
SubSectionNames id	INT	Sim	Não	Sim	
interperptation	TINYINT	Não	Não	Não	Elemento contém interpretação clínica: 0 – Desvio-padrão; 1 – Pontuação escalar; 2 – Não cognitiva; NULL – Não contém interpretação clínica.

Tabela A.57 - Tabela "CognitiveProfile" na base de dados.

Tabela "CognitiveProfile": Armazena informação sobre o perfil cognitivo do paciente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Patients id	INT	Sim	Não	Sim	
domGeneralCognition	DECIMAL(5,2)	Sim	Não	Não	
domOrientation	DECIMAL(5,2)	Sim	Não	Não	
domMemory	DECIMAL(5,2)	Sim	Não	Não	
domExecutiveFunctions	DECIMAL(5,2)	Sim	Não	Não	
domLanguage	DECIMAL(5,2)	Sim	Não	Não	
domAttention	DECIMAL(5,2)	Sim	Não	Não	
dateCreated	DATETIME	Sim	Não	Não	

Tabela A.58 - Tabela "GDS_15" na base de dados.

Tabela "GDS_15": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
score15	INT	Sim	Não	Não	Campo GDS15 Min: 0 Max: 15

Tabela A.59 - Tabela "MaritalStatus" na base de dados.

Tabela "MaritalStatus": Armazena informação sobre o estado civil. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
maritalStatus_name	VARCHAR(45)	Sim	Não	Não	

Tabela A.60 - Tabela "Sessions_Assessment" na base de dados.

Tabela "Sessions_Assessment": Armazena informações sobre as sessões de avaliação neuropsicológica.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
dateOfAssessment	DATETIME	Sim	Não	Não	

Tabela A.61 - Tabela "ActivitiesSettings" na base de dados.

Tabela "ActivitiesSettings": Armazena as configurações existentes disponíveis em cada actividade cognitiva. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Activities_id	INT	Sim	Não	Sim	
settingName	VARCHAR(45)	Sim	Não	Não	
type	VARCHAR(45)	Sim	Não	Não	
name	VARCHAR(45)	Sim	Não	Não	

Tabela A.62 - Tabela "*ActivitiesSettingsIterations*" na base de dados.

Tabela " <i>ActivitiesSettingsIterations</i> ": Armazena a informação das definições das atividades atribuídas às sessões..					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
ActivitiesSettings_id	INT	Sim	Não	Sim	
ActivitiesIterations_id	INT	Sim	Não	Sim	
value	VARCHAR(50)	Sim	Não	Não	

Tabela A.63 - Tabela "*QOLIBRI*" na base de dados.

Tabela " <i>QOLIBRI</i> ": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
total	DECIMAL(5,2)	Sim	Não	Não	Campo Ensaio da evocação imediata Min: 0 Max: 15
cognitive	DECIMAL(5,2)	Sim	Não	Não	Campo Resultado combinado do reconhecimento Min: 0 Max: 30
self	DECIMAL(5,2)	Sim	Não	Não	
avds	DECIMAL(5,2)	Sim	Não	Não	
socialRelationships	DECIMAL(5,2)	Sim	Não	Não	
emotions	DECIMAL(5,2)	Sim	Não	Não	
physicalCondition	DECIMAL(5,2)	Sim	Não	Não	

Tabela A.64 - Tabela "*BDI_IP*" na base de dados.

Tabela " <i>BDI_IP</i> ": Resultados durante a aplicação do NPA. As entradas obrigatórias são sempre preenchidas pelo profissional de saúde.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
BDIII	INT	Sim	Não	Não	Campo BDI-II Min: 0 Max: 63

Tabela A.65 - Tabela "*ProfessionalSituation*" na base de dados.

Tabela " <i>ProfessionalSituation</i> ": Armazena informação sobre a situação profissional do paciente. Esta informação é introduzida manualmente.						
Atributos						
Nome	Tipo	Não nulo	PK	FK	Default	Significado
id	INT	Sim	Sim	Não		
name	VARCHAR(45)	Sim	Não	Não		

Tabela A.66 - Tabela "*NormativeDataCognitive*" na base de dados.

Tabela " <i>NormativeDataCognitive</i> ": Armazena informação sobre as interpretações clínicas e dados normativos cognitivos associados a um campo NPA a ser introduzido.						
Atributos						
Nome	Tipo	Não nulo	PK	FK	Default	Significado
id	INT	Sim	Sim	Não		
NPAAElementDescriptor_id	INT	Sim	Não	Sim		
AssessmentByInstruments_id	INT	Sim	Não	Sim		
PatternDeviationInterpretation_id	INT	Não	Não	Sim		
ScalarScoreInterpretation_id	INT	Não	Não	Sim		
PatternDeviationScalarScore_id	INT	Não	Não	Sim		

Tabela A.67 - Tabela "*ScalarScoreInterpretation*" na base de dados.

Tabela " <i>ScalarScoreInterpretation</i> ": Interpretação cognitiva clínica: pontuações escalares. Esta tabela contém as escalas, nomes e pontuações escalares. Esta informação é introduzida manualmente.						
Atributos						
Nome	Tipo	Não nulo	PK	FK	Default	Significado
id	INT	Sim	Sim	Não		
levelOfInterpretation	VARCHAR(45)	Sim	Não	Não		
scalarScores	VARCHAR(45)	Sim	Não	Não		
scale	INT	Sim	Não	Não		

Tabela A.68 - Tabela "*PatternDeviationInterpertation*" na base de dados.

Tabela " <i>PatternDeviationInterpertation</i> ": Interpretação clínica cognitiva: média +- desvio padrão Esta tabela contém as escalas, os nomes e o desvio médio +-padrão. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
levelOfInterpretation	VARCHAR(45)	Sim	Não	Não	
patternDeviationAverage	VARCHAR(45)	Sim	Não	Não	
scale	INT	Sim	Não	Não	

Tabela A.69 - Tabela "*CutoffScore*" na base de dados.

Tabela " <i>CutoffScore</i> ": Armazena informação sobre as interpretações clínicas não cognitivas associadas a um campo NPA a ser introduzido.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
NPAElementDescriptor_id	INT	Sim	Não	Sim	
AssessmentByInstruments_id	INT	Sim	Não	Sim	
NoCognitiveInterpertation_id	INT	Sim	Não	Sim	

Tabela A.70 - Tabela "*NoCognitiveInterpertation*" na base de dados.

Tabela " <i>NoCognitiveInterpertation</i> ": Interpretação clínica não cognitiva Esta tabela contém as escalas e os nomes. Esta informação é introduzida manualmente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
levelOfInterpretation	VARCHAR(45)	Sim	Não	Não	
scale	VARCHAR(45)	Sim	Não	Não	
NPA_id	INT	Sim	Não	Sim	

Tabela A.71 - Tabela "PatternDeviationScalarScore" na base de dados.

Tabela "PatternDeviationScalarScore": Armazena informação sobre a média +- desvio padrão ou pontuação escalar, introduzida pelo profissional de saúde dependendo do campo NPA introduzido.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
average	DECIMAL(5,2)	Não	Não	Não	
patternDeviation	DECIMAL(5,2)	Não	Não	Não	
scalarScore	INT	Não	Não	Não	

Tabela A.72 - Tabela "InterperationClinical" na base de dados.

Tabela "InterperationClinical": Armazena informação sobre a interpretação clínica do paciente.					
Atributos					
Nome	Tipo	Não nulo	PK	FK	Significado
id	INT	Sim	Sim	Não	
Patients_id	INT	Sim	Não	Sim	
domGeneralCognition	INT	Sim	Não	Não	
domOrientation	INT	Sim	Não	Não	
domMemory	INT	Sim	Não	Não	
domExecutiveFunctions	INT	Sim	Não	Não	
domLanguage	INT	Sim	Não	Não	
domAttention	INT	Sim	Não	Não	
dateCreated	DATETIME	Sim	Não	Não	

Anexo B – Funcionalidades

The screenshot shows the 'Informações do paciente - José Manuel Cunha' page. The left sidebar contains navigation options: Pacientes (Novo paciente, Lista de pacientes), Programas de treino (Programas em curso, Programas finalizados), and Geral (Consultar perfil, Logout). The main content area has tabs for 'Dados do paciente', 'Informação clínica', 'Avaliações cognitivas', 'Programas de treino', and 'Perfil cognitivo'. The 'Dados do paciente' tab is active, displaying a form with the following fields: Nome completo* (José Manuel Cunha), Email* (josemanuel@mail.pt), Contato telefónico* (929222737), Morada (Funchal), Profissão* (Manobrador), Situação profissional* (Empregado), Data de nascimento* (05/10/1988), Sexo* (Masculino), Lateralidade* (Esquerdino), Escolaridade* (15), Agregado familiar* (Pa), and Estado civil* (Viúvo). At the bottom of the form are buttons for 'Guardar alterações' and 'Cancelar alterações'.

Fig. B.1 – Funcionalidade de edição das informações sociodemográficas do paciente.

The screenshot shows the 'Informações do paciente - José Manuel Cunha' page with the 'Informação clínica' tab active. The form includes a dropdown for 'Tipo de informação clínica*' (Terapêutica farmacológica) and a text area for 'Descrição*' containing the text: 'O paciente atualmente é medicado com ... 3 vezes por semana.' Below the form is a 'Registrar informação clínica' button. Below the form is a table with a search bar and the following data:

Data de registo ↓	Tipo de informação	Descrição
2022-10-07	Diagnóstico/história clínica	Sofreu um AVC - Acidente vascular cerebral no dia 11 de Maio de 2019.

Fig. B.2 – Funcionalidade de registo de uma informação clínica.

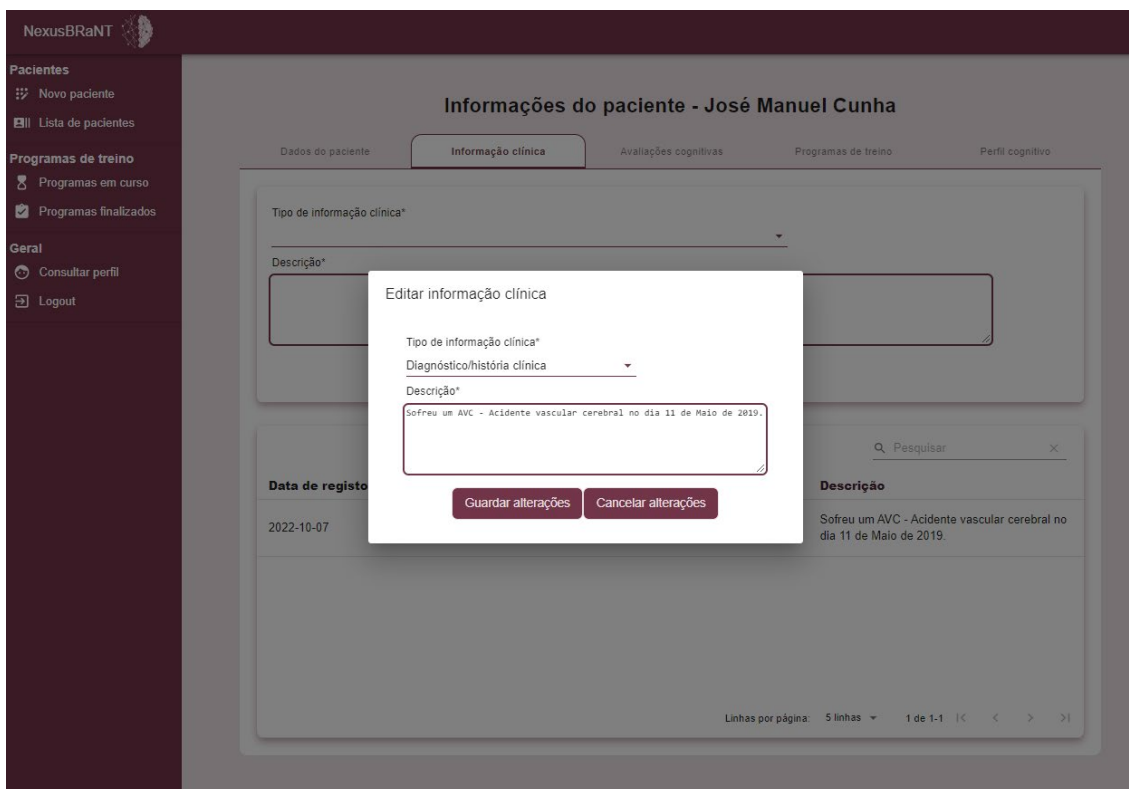


Fig. B.3 – Funcionalidade de edição de uma informação clínica.

Interpretação clínica - Cognitiva			
Médias e desvios-padrão			
id	Níveis de interpretação	Desvio-padrão da média	Escala
1	Alteração muito severa/extrema	<-3 DP ou mais	1
2	Alteração severa	<-2 DP a -3 DP	2
3	Alteração moderada	<-1.5 DP a -2 DP	3
4	Alteração ligeira	<-1 DP a -1.5 DP	4
5	Médio	-1 DP a 1 DP	5
6	Médio superior	>1 DP a 1.5 DP	6
7	Superior	>1.5 DP a 2 DP	7
8	Muito superior	>2 DP a 3 DP	8
9	Extremamente superior	>3 DP ou mais	9
Instrumentos de avaliação			
Figura Complexa de Rey			
MOCA			
Toulouse-Piéron			
Pontuações escalares			
id	Níveis de interpretação	Pontuações escalares	Escala
1	Extremamente inferior	1-2	1
2	Muito inferior	3-4	2
3	Inferior	5-6	3
4	Médio inferior	7-8	4
5	Médio	9-11	5
6	Médio superior	12-13	6
7	Superior	14-15	7
8	Muito superior	16-17	8
9	Extremamente elevado	18-19	9
Instrumentos de avaliação			
Fluências Verbais Semânticas e Fonéticas			
WAIS			
WMS			

a)

Interpretação clínica - Não cognitiva			
CDR			
id	Níveis de interpretação	Escala	NPA
1	Nenhuma	0	7
2	Suspeita	0,5	7
3	Ligeira	1	7
4	Moderada	2	7
5	Grave	3	7
BDI-II			
id	Níveis de interpretação	Escala	NPA
6	Depressão mínima	0-11	20
7	Depressão ligeira	12-18	20
8	Depressão moderada	19-24	20
9	Depressão severa	>25	20
GDS-15			
id	Níveis de interpretação	Escala	NPA
10	Normal	0-4	4
11	Depressão ligeira	5-8	4
12	Depressão moderada	9-11	4
13	Depressão severa	12-15	4
GDS-30			
id	Níveis de interpretação	Escala	NPA
14	Sintomatologia normal	≤10	3
15	Depressão ligeira	11-20	3
16	Depressão severa	≥21	3

b)

Fig. B.4 - Interpretação clínica: a) cognitiva; b) não cognitiva.

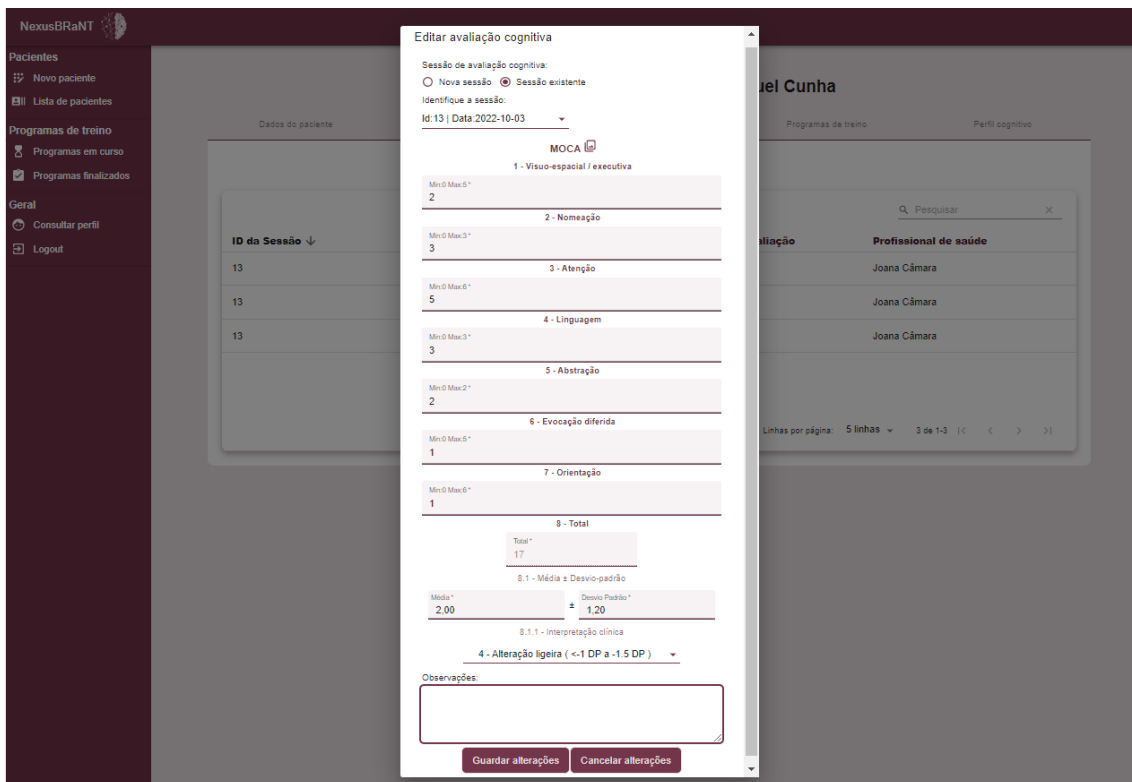


Fig. B.5 – Funcionalidade de edição da avaliação neuropsicológica selecionada.

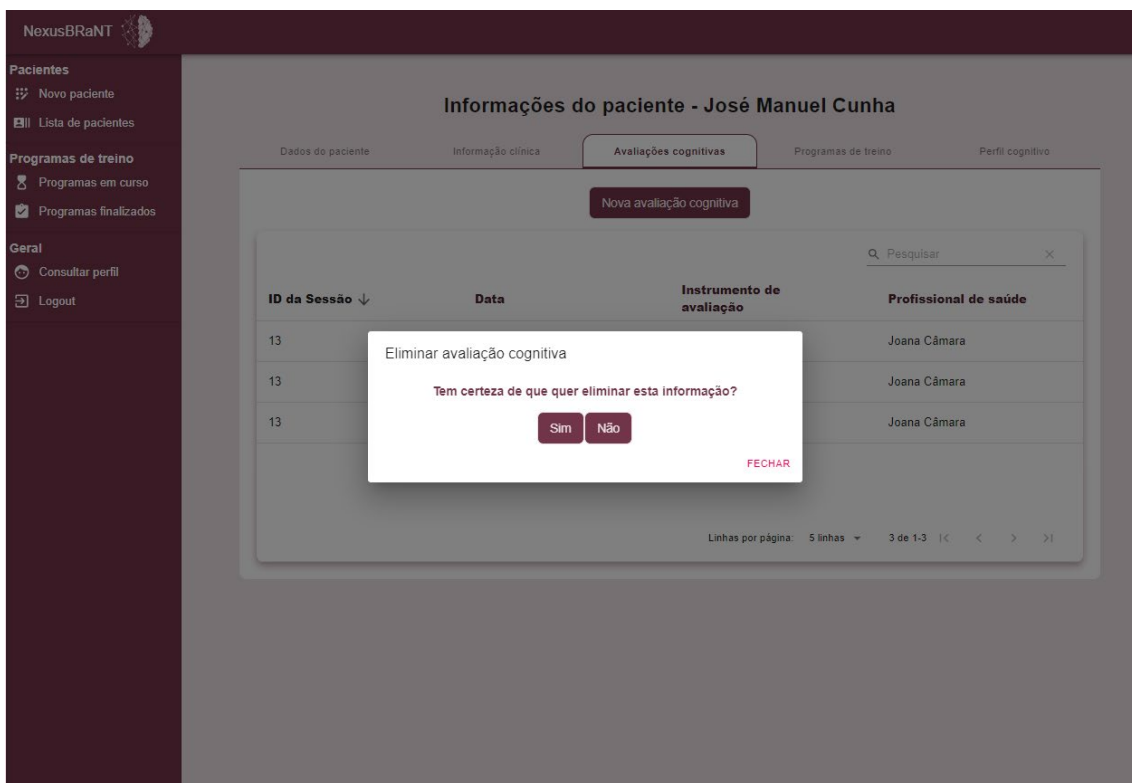


Fig. B.6 – Funcionalidade de eliminação da avaliação neuropsicológica selecionada.

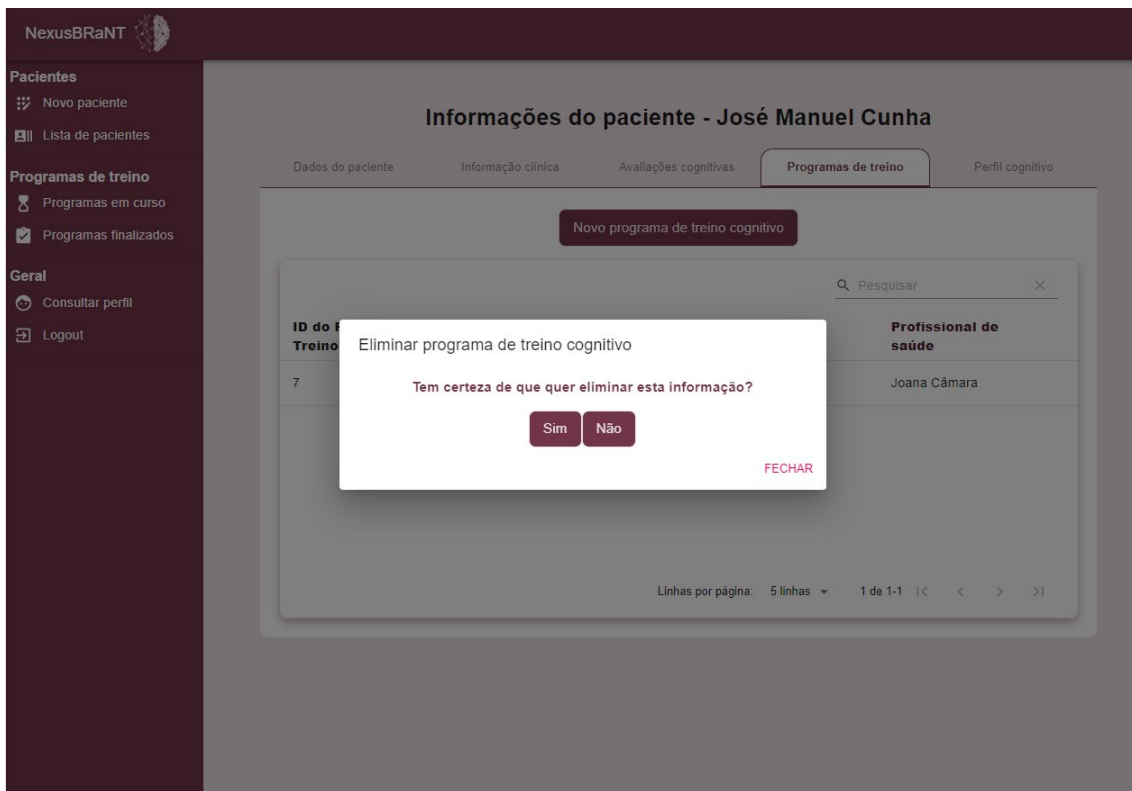


Fig. B.7 – Funcionalidade de eliminação do programa de treino cognitivo selecionado.

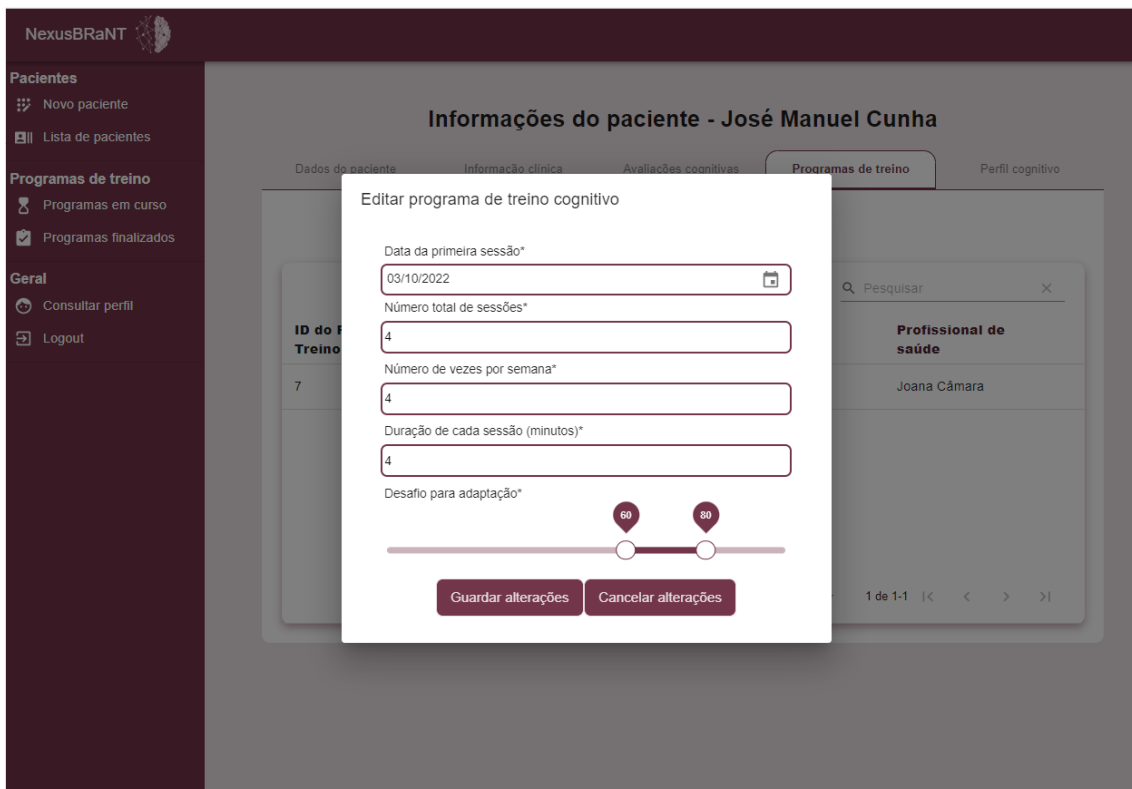


Fig. B.8 – Funcionalidade de edição de um programa de treino cognitivos selecionado.

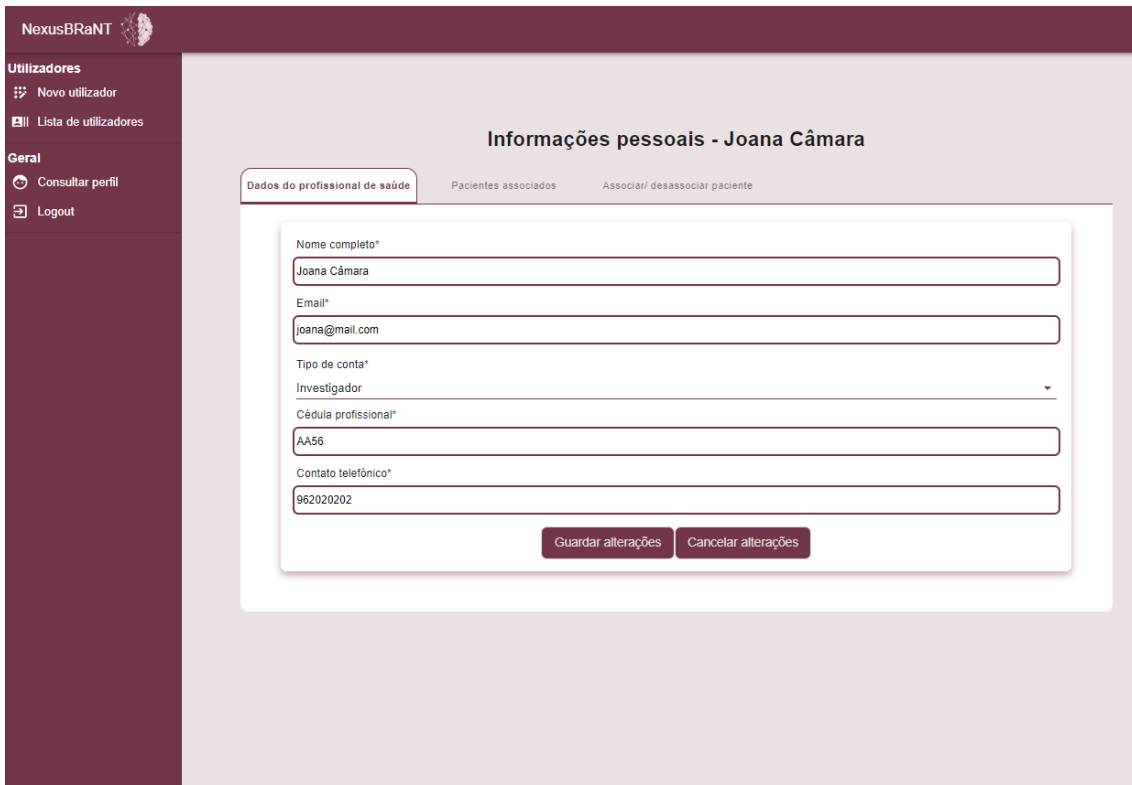
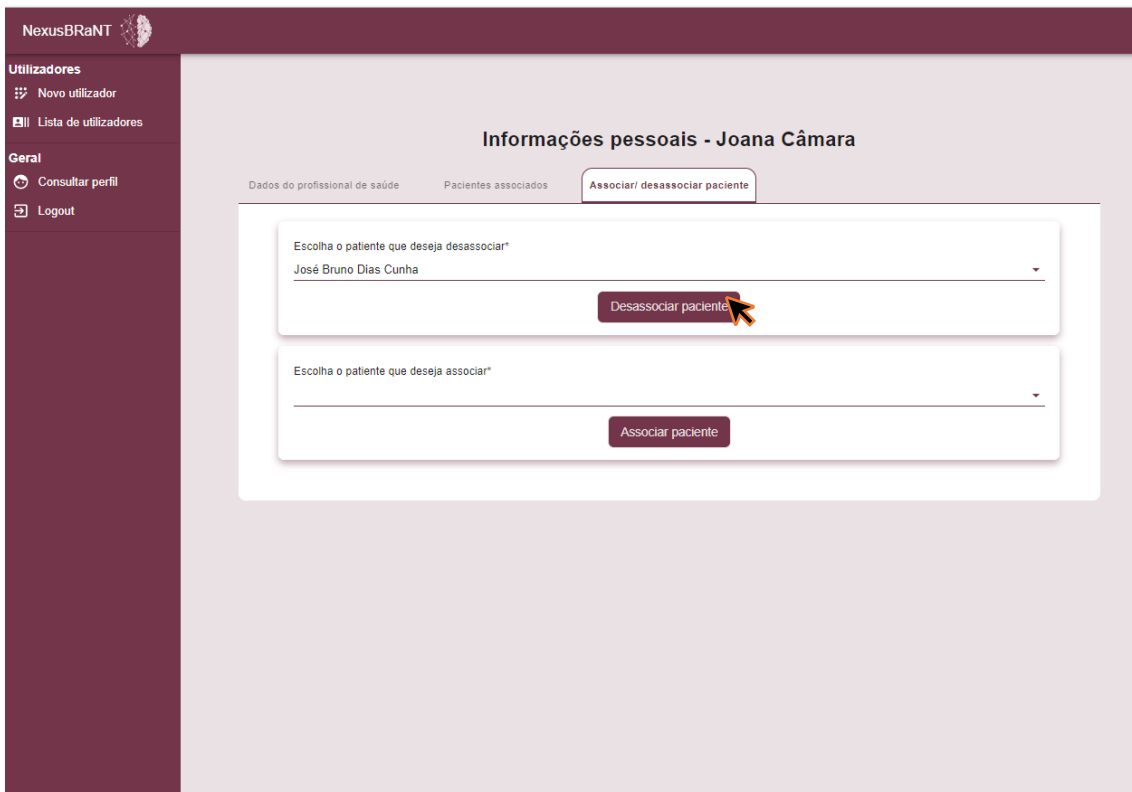
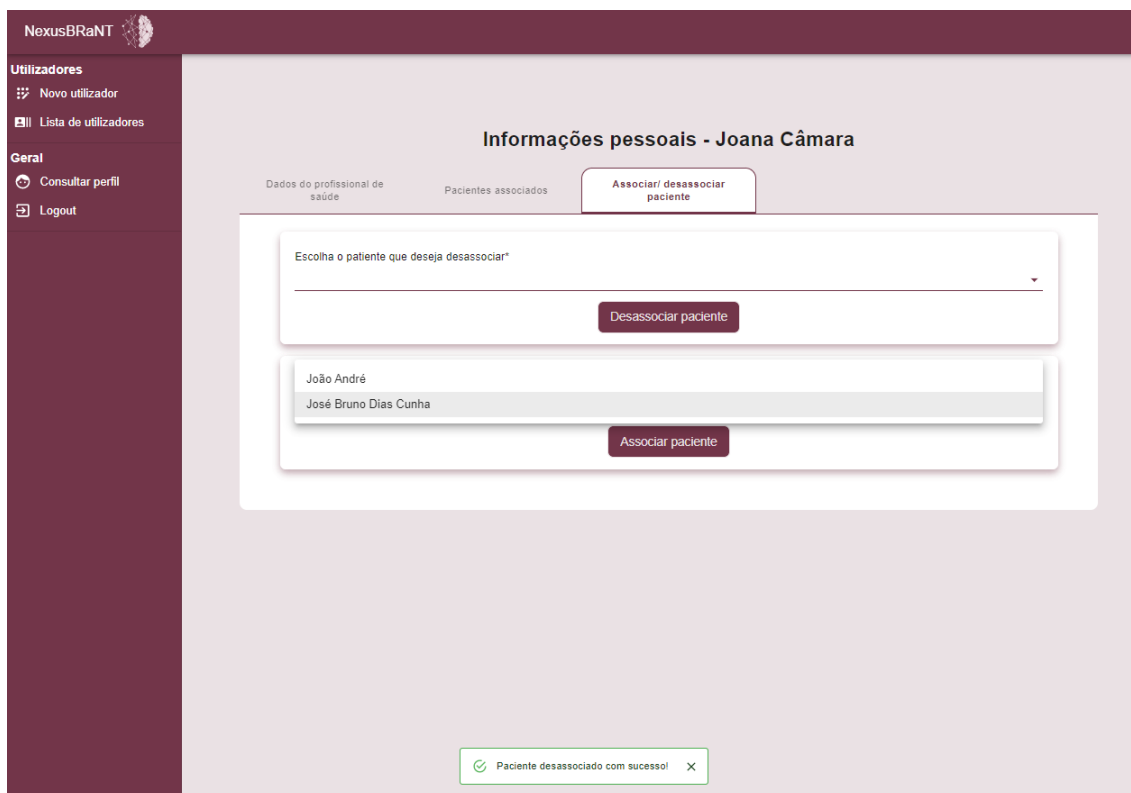


Fig. B.9 – Funcionalidade de edição dos dados do PS selecionado.



a)



b)

Fig. B.10 – Associar/desassociar o paciente José Bruno: a) seleção do paciente; b) após ter desassociado o paciente, o mesmo aparece na seleção de associar e mensagem de sucesso da desassociação.

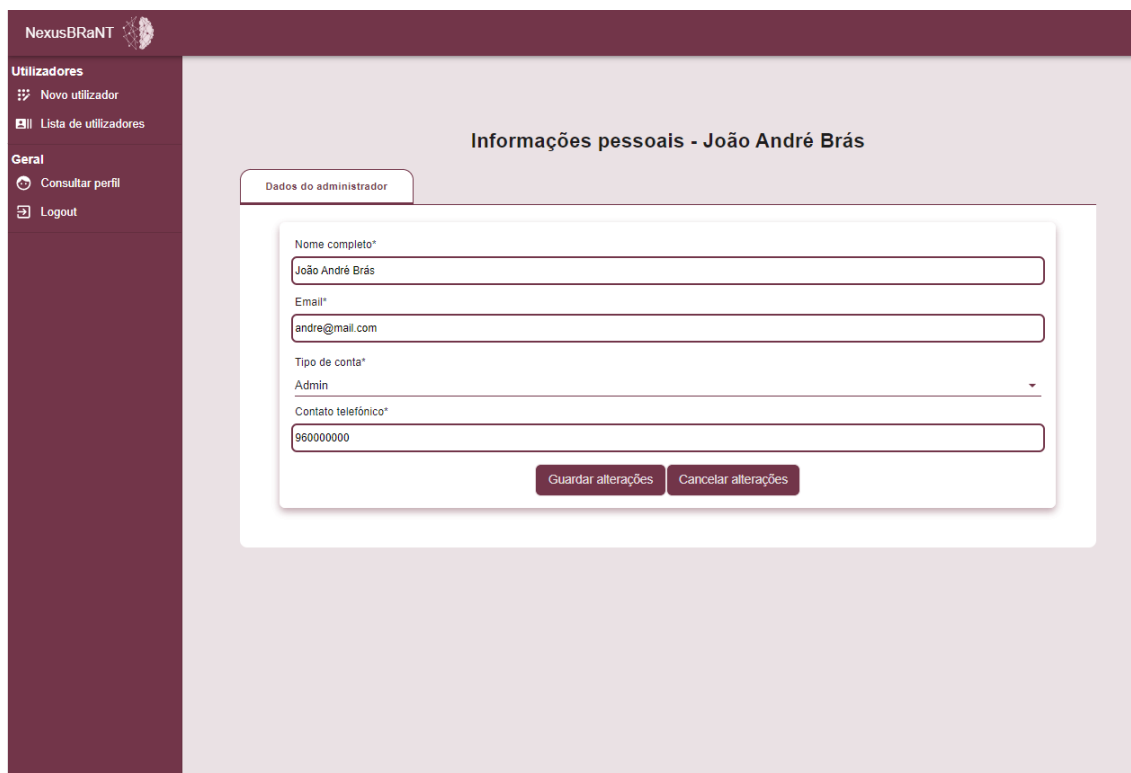


Fig. B.11 – Funcionalidade de edição dos dados do administrador autenticado.

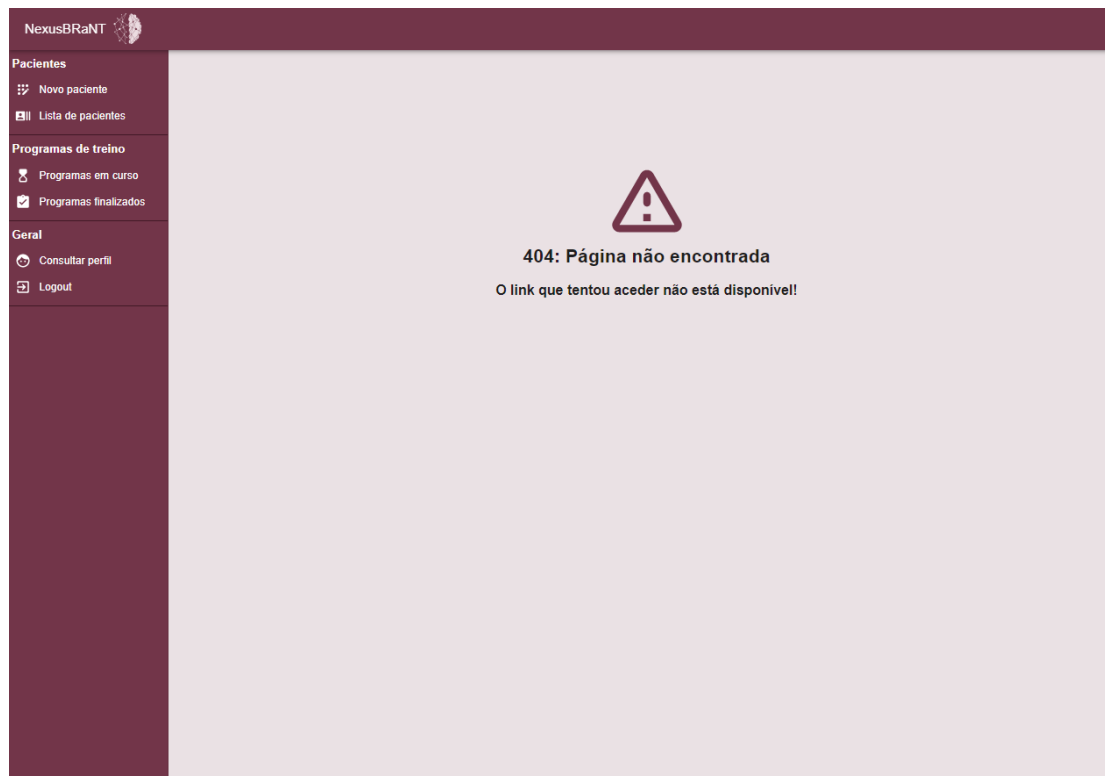


Fig. B.12 – Funcionalidade da página não encontrada.

Anexo C – Avaliação do sistema



a)

Introdução

BRaNT

- Objetivo do projeto BRaNT
 - Criar novas ferramentas tecnológicas para a reabilitação cognitiva domiciliária e fornecer soluções para sistemas de saúde mais sustentáveis.
- Objetivo da ferramenta NeuroAreh@tx
 - Criar um perfil cognitivo que otimiza a adaptação da prescrição através de inteligência artificial;
 - Fornecer ao terapeuta várias sugestões para uma reabilitação cognitiva altamente personalizada e adaptativa através de atividades em realidade virtual baseadas na simulação de vida diária.

2 de 5

b)

Introdução

NexusBRaNT

- O que é?
 - Plataforma online para aceder ao *backoffice* do projeto BRaNT.
- Público-alvo:
 - Profissionais de saúde (psicólogos, neuropsicólogos e neurologistas, entre outros);
 - Realizam reabilitação cognitiva e trabalham quer em contexto hospitalar como em clínicas privadas.

3 de 5

c)

Introdução

NexusBRaNT

Principais funcionalidades do NexusBRaNT

Gestão de utentes	Gestão de avaliações neuropsicológicas	Gestão de treinos cognitivos
<ul style="list-style-type: none">• Adicionar um novo utente;• Adicionar e consultar a informação clínica;• Visualizar a evolução do utente;• Calcular a performance absoluta;• Calcular a interpretação clínica.	<ul style="list-style-type: none">• Criar a sessão de avaliação;• Criar a avaliação associada a um instrumento de avaliação;• Preencher a interpretação clínica;• Visualizar os resultados de avaliações.	<ul style="list-style-type: none">• Criar um treino cognitivo;• Visualizar um programa de treino;• Visualizar as sessões de treino e progresso do paciente.

4 de 5

d)

Obrigada pela atenção!

4 de 5

e)

Fig. C.1 – PowerPoint de testes de usabilidade: a) Capa; b) BRaNT; c) NexusBRaNT (definição e público-alvo); d) NexusBRaNT (principais funcionalidades); e) Agradecimento.

NexusBRaNT - System Usability Scale

Principal researcher: [Érica Cunha, Universidade da Madeira, 2033617@student.uma.pt]
Secondary

researcher: [Docente Yuri Almeida, Universidade da Madeira, nosferatus.sos@gmail.com]
Secondary researcher: [Docente Filipe Quintal, Universidade da Madeira, filipe.quintal@staff.uma.pt]

Purpose of the Study
As part of the master's thesis of the Computer Engineering course, this questionnaire is carried out to quantitatively measure the usability of the NexusBRaNT system.

Procedure
By agreeing to participate in this study, you will answer a questionnaire with an approximate completion time of 3 to 5 minutes.



Inclusion Criteria
You will be considered eligible to participate in this study if you are a healthcare professional or healthcare student.

Risks
There are no risks to your participation.

Benefits
This study will not have an immediate benefit to you, but the information collected will allow us to contribute to the development of technologies to support cognitive rehabilitation health professionals.



Compensation & Costs
There will be no compensation. There is no cost to participate in this study.

Confidentiality
Confidentiality of data will be maintained in the following ways: No identifiable data will be collected. The data collected will be used for research purposes only.

 pipasmile@gmail.com (não partilhado) [Mudar de conta](#) 

a)

NexusBRaNT - System Usability Scale

 (não partilhado) [Mudar de conta](#) 

*Obrigatório

Consent and authorization

Rights
Your participation is voluntary. You are free to discontinue your participation at any time. Refusal to participate or discontinuation of participation will not result in any penalty, or loss of any benefits or rights. The principal investigator may make a reasoned decision to discontinue your participation in this study. Should this situation arise, it will not result in any penalty, or loss of any benefits or rights.

Questions and Contacts
If you have questions about this study at any time, you may contact the responsible investigators through the contacts listed above.

Voluntary Informed Consent
By agreeing to participate in this study, you confirm that you have read the above information about it, and that all your questions have been answered.

Please indicate whether you agree to participate in this study *

I agree to participate in this study.

I do not agree to participate in this study.



Anterior
Seguinte

Limpar formulário

Página 2 de 4

b)

NexusBRaNT - System Usability Scale

 (não partilhado) [Mudar de conta](#)  Rascunho guardado

*Obrigatório

Demographics of the participants

Demographic data will be collected for study purpose only. These data will be kept anonymous and used for discussion of results.

User ID

0 _____

Age *

0 _____

Nationality *

. _____

Gender *

Female

Male

Professional status *

Student



Health professional

Health professional in the BRaNT project

Anterior
SeguintePágina 3 de 4
Limpar formulário

c)

NexusBRaNT - System Usability Scale

 pipasmile@gmail.com (não partilhado) [Mudar de conta](#) 

*Obrigatório

System Usability Scale

The System Usability Scale (SUS) provides a "quick and dirty", reliable tool for measuring the usability. It consists of a 10 item questionnaire with five response options for respondents; from Strongly agree to Strongly disagree.

1. I think that I would like to use this system frequently *

1 2 3 4 5

Strongly disagree Strongly agree

2. I found the system unnecessarily complex *

1 2 3 4 5

Strongly disagree Strongly agree

3. I thought the system was easy to use *

1 2 3 4 5

Strongly disagree Strongly agree

4. I think that I would need the support of a technical person to be able to use this system *

1 2 3 4 5

Strongly disagree Strongly agree

5. I found the various functions in this system were well integrated *

1 2 3 4 5

Strongly disagree Strongly agree

6. I thought there was too much inconsistency in this system *

1 2 3 4 5

Strongly disagree Strongly agree

7. I would imagine that most people would learn to use this system very quickly *

1 2 3 4 5

Strongly disagree Strongly agree

8. I found the system very cumbersome to use *

1 2 3 4 5

Strongly disagree Strongly agree

9. I felt very confident using the system *

1 2 3 4 5

Strongly disagree Strongly agree

10. I needed to learn a lot of things before I could get going with this system *

1 2 3 4 5

Strongly disagree Strongly agree

Anterior **Enviar** ▬ Página 4 de 4 Limpar formulário

Nunca envie palavras-passe através dos Google Forms.
Este conteúdo não foi criado nem aprovado pela Google. [Denunciar abuso](#) - [Termos de Utilização](#) - [Política de privacidade](#)

Google Formulários

d)

Fig. C.2 – Questionário SUS: a) informações acerca do estudo realizado; b) consentimento e autorização; c) dados demográficos do paciente; d) questões do SUS.