

Mestrado em Engenharia Informática
Estágio

Relatório Final

Serious Funny Social Game for Brain Gym

David Carvalho

demc@student.dei.uc.pt

Orientadores:

Prof. Dr. Luis Macedo (DEI)

Eng. Nuno Machado (Cnotinfor)

Data: 28 de Agosto de 2014



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

O envelhecimento traz reduções significativas nas capacidades cognitivas do ser humano. O treino mental, através de atividades como pequenos jogos de memorização, lógica e raciocínio, é uma forma eficaz de impedir a senilidade precoce e, pode mesmo atrasar o desenvolvimento da doença de Alzheimer. No entanto, o isolamento das pessoas de 3ª idade cria dificuldades na criação de ambientes onde haja estimulação cerebral e interação social, arma importante no desenvolvimento cognitivo.

Com a generalização do acesso à internet e dos computadores, torna-se possível um contacto mais próximo da pessoa idosa com a família, com amigos e, com o seu médico. Se, aliado a este contato, conseguirmos integrar jogos atrativos que estimulem as capacidades mentais e retornem resultados mensuráveis que sirvam de diagnóstico progressivo para o seu psicogeriatra ou outro médico especialista na saúde mental do idoso, conseguiremos ter uma ferramenta que melhora substancialmente a vida da população de 3ª idade.

O trabalho aqui apresentado representa o arranque de um projeto complexo, que visa a criação de uma plataforma integradora de um ginásio cerebral através de pequenos jogos e, que contenha também uma forte componente de socialização. Pretende-se assim aumentar ainda mais o estímulo da mente e aproximar o idoso da família e amigos, criando ao mesmo tempo dados clínicos evolutivos, para uma perceção fácil do desenvolvimento das capacidades cognitivas do idoso e consequente aconselhamento.

Este trabalho é o resultado de um estágio curricular, no âmbito do Mestrado em Engenharia Informática pela Universidade de Coimbra, realizado na Cnotinfor, empresa proponente do estágio.

Palavras-Chave

"active brain" "serious funny social game for brain gym" "social game" "ginásio cerebral" "aplicações para idosos" "combate à senilidade" "software de geriatria" "geriatria remota" "ginásio cerebral" "estimulação cognitiva"

Índice

Capítulo 1 Introdução	1
1.1. Motivação.....	1
1.2. O Projeto ActiveBrain	1
1.3. Cnotinfor (Empresa proponente)	2
1.4. Estágio.....	2
1.6. Estrutura do documento	3
Capítulo 2 Estado da Arte.....	5
2.1. Estudo do público-alvo	5
2.2. Combate à senescência e senilidade nos dias de hoje	5
2.3. Plataformas “Ginásio Cerebral”	6
2.3.1. Lumosity (lumosity.com) ^[3]	6
2.3.2. Fit Brains Trainer (fitbrains.com) ^[4]	7
2.3.3. Cérebro Melhor (cerebromelhor.com.br) ^[5]	7
2.3.4. Cognifit (cognifit.com) ^[6]	7
2.3.5. Comparativo	7
2.2. Conclusão.....	8
Capítulo 3 Especificação	9
3.1. Descrição Geral.....	9
3.2. Caracterização do utilizador.....	10
3.3. Dispositivos	11
3.4. Análise de Requisitos	11
3.4.1. Requisitos Funcionais	11
3.4.2. Requisitos não-funcionais	13
3.4.2.1. Persistência de dados	13
3.4.2.2. Sincronização de dados	2
Capítulo 4 Arquitetura do Sistema	3
4.1. Visão geral.....	3
4.2. Estrutura de dados.....	4
4.2.1. Estrutura de dados locais.....	4
4.2.2. Estrutura de dados remotos.....	4
4.3. Estrutura de Classes	6

4.4. API REST	6
4.5. Fluxo de Dados.....	8
4.6. Lógica do jogo.....	9
Capítulo 5 Tecnologias	11
5.1. Framework / Linguagem de desenvolvimento da aplicação	11
5.1.1. Análise de linguagens/frameworks existentes	11
5.1.2. Comparativo	12
5.1.3. A escolha da framework Cocos2d-x.....	12
5.1.4. TopQX	13
5.2. Base de dados local.....	13
5.3. Base de dados remota.....	13
5.4. API REST	14
Capítulo 6 Desenvolvimento	15
6.1. Calendarização das tarefas.....	15
6.2. Metodologia.....	17
6.3. Gestão de riscos	17
6.4. Controlo de versões e backup de dados.....	20
6.5. Documentação	21
6.6. Progresso de desenvolvimento e desvios ao planeamento	21
6.7. Outras tarefas envolvendo o estagiário	22
6.7.1. Implementação de duas novas classes na biblioteca TopQX.....	22
6.7.2. Microsoft Developer Tech Refresh.....	23
6.7.3. Escrita de artigo para revista da empresa	23
Capítulo 7 Conclusões e Trabalho Futuro.....	25
Referências	27
ANEXO A – Requisitos Funcionais da aplicação	29
ANEXO B – Diagramas de Gantt da calendarização das várias tarefas	41
ANEXO C – Exemplos de especificação da API REST	47
ANEXO D – Exemplos de interfaces implementadas.....	51

Lista de Figuras

Figura 1 - Representação da Arquitetura do Sistema.....	3
Figura 2 - Diagrama ER de dados locais, do ActiveBrain	4
Figura 3 - Diagrama ER de dados remotos, do ActiveBrain.....	5
Figura 4 - Estrutura de classes implementada em MVC.....	6
Figura 5 - Diagrama de Sequência da chamada de validação de Login	8
Figura 6 - Esboço de um mapa com o posicionamento dos pontos definido em porcentagem	9
Figura 7 - Exemplo da definição de um mapa de jogo em XML, com sua hierarquia e atributos	9
Figura 8 - Arquitetura da Framework Cocos2d-x [8].....	13
Figura 9 - Diagrama de Gantt relativo ao 1º semestre de Estágio	15
Figura 10 - Diagrama de Gantt previsto para o 2º semestre de Estágio, no período de 27 de Janeiro a 4 de Julho.....	16
Figura 11 - Diagrama de Gantt efetivo do 2º semestre de Estágio, no período de 27 de Janeiro a 28 de Agosto	16
Figura 12- Diagrama de Casos de Uso para o “Painel de Autenticação”	29
Figura 13 - Diagrama de Casos de Uso para o “Menu Principal”	31
Figura 14 - Diagrama de Casos de Uso para “Perfil de Utilizador”	32
Figura 15 - Diagrama de Casos de Uso para o “Painel de Jogos”	34
Figura 16 - Diagrama de Casos de Uso para “Ver Histórias”	35
Figura 17 - Diagrama de Casos de Uso para o “Ver notícias”	36
Figura 18 - Fluxo do jogo “Qual o melhor caminho”	38

Lista de Tabelas

Tabela 1 - Comparativo entre aplicações de treino cerebral existentes	7
Tabela 2 - Lista de requisitos funcionais da aplicação	13
Tabela 3 - Comparativo entre frameworks de implementação multidispositivo	12

Lista de Acrónimos

DEI	Departamento de Engenharia Informática da Universidade de Coimbra
HTML	HyperText Markup Language
IDE	Integrated Development Environment
API	Application Programming Interface
PHP	Hypertext Preprocessor
DBMS	Database Management System
HTTP	Hypertext Transfer Protocol
XML	Extensible Markup Language
JSON	Javascript Object Notation
BD	Base de Dados
PC	Personal Computer
2D	2 dimensões. I.e., abstração de profundidade
3D	3 dimensões. I.e., uso de técnicas de simulação de profundidade
UML	Unified Modeling Language
API	Application Programming Interface
REST	Representational State Transfer

Capítulo 1

Introdução

1.1. Motivação

O enfraquecimento das capacidades cognitivas é uma condição natural do ser humano e, apesar das capacidades mentais decaírem de forma diferente dentro da mesma faixa etária, a demência é um conjunto de doenças quase inevitável com o envelhecimento. Este fato acontece principalmente depois dos 65 anos (5% da população sofre de demência nesta altura), sendo que a sua evolução tem um perfil exponencial com o avançar da idade^[1]. Dado que o envelhecimento da população mundial tem vindo a acentuar-se, estima-se que o número de idosos afetados pela demência praticamente duplique em 20 anos.^[16]

Existem tendências claras das áreas cognitivas com maior degradação. A atenção, a memória, a percepção espacial, as funções executivas e a velocidade de processamento são as capacidades mais afetadas pelo envelhecimento.^[2]

Associado à avançada idade, infelizmente, ouvimos muitas vezes falar de solidão. Este é um fato real e com tendência crescente, devido a inúmeros fatores como a migração dos filhos, o envelhecimento da população em geral, as dificuldades económicas que muitas vezes impedem familiares de estarem juntos mais vezes, etc.

1.2. O Projeto ActiveBrain

Foi a pensar nos idosos, no declínio das funções mentais e no isolamento, que surgiu o projeto ActiveBrain, nome comercial desta plataforma tecnológica, que visa o estímulo da mente dos idosos através de 3 componentes chave: Jogos de treino cognitivo, partilha de histórias entre utilizadores e divulgação de notícias/informações de interesse para o utilizador.

Inicialmente este projeto arrancou com maior foco no jogo, mas depois de uma análise a outros produtos e ao aprofundamento dos objetivos chave, nomeadamente o estímulo cerebral e a socialização, a componente jogo passou a ser apenas uma de 3 componentes, como veremos mais à frente. Este é ainda assim, a componente mais importante, pois visa o uso de jogos atrativos e sociais que estimulem o cérebro do idoso (através de desafios de lógica, raciocínio, atenção, etc.), combatendo assim a senilidade mas também a solidão. Pretende-se que estes jogos promovam a partilha de experiência e a partilha de jogabilidade (dando a oportunidade de 2 pessoas se ajudarem mutuamente ou jogarem um contra o outro), ao mesmo tempo que permitirá uma conversação contínua entre jogadores.

Além destas 3 áreas de intervenção direta dos utilizadores, no combate ao sedentarismo mental e às doenças degenerativas cerebrais e, com socialização ativa, fator diferenciador em relação a plataformas já existentes, o ActiveBrain pretende introduzir outra ferramenta inovadora neste tipo de aplicações: Produção de dados médicos e comunicação com o profissional de medicina que acompanha o idoso. Isto permite que um psicogeriatra, terapeuta de reabilitação ou outro profissional de saúde, e até mesmo a família, possam receber dados mensuráveis provenientes da jogabilidade, para poder avaliar o progresso das suas capacidades cognitivas e portanto, da sua doença degenerativa. Assim permite um acompanhamento mais constante

e menos custoso para o idoso, que muitas vezes vê-se com dificuldades em se deslocar a um centro de saúde devido à sua situação geográfica ou financeira.

Os dispositivos acessíveis ao Público-Alvo deste projeto nem sempre são os mesmos (PC's, tablets, Mac, etc.) e, visto que se pretende uma plataforma com treino progressivo e portanto, com registo do estado evolutivo de cada pessoa, não é viável o uso de uma aplicação que guarde estes dados unicamente na máquina onde corre. Pretende-se que os dados sejam centralizados num centro de dados, para que possam ser lidos tanto pelo paciente em qualquer dispositivo (ex.: no PC do lar que frequenta, no seu tablet), mas também pelo profissional de saúde que o acompanha, através de uma aplicação de acompanhamento (portal web). Este projeto visa assim, a implementação de uma plataforma interoperável e com sincronização de dados.

Ao falarmos da 3ª idade, falamos da faixa etária mais velha da sociedade, que no geral tem acesso limitado ou não tem mesmo acesso à tecnologia, e muitas vezes opõe-se ao seu uso. A maioria por isso, não está familiarizada com alguns dispositivos, pelo que implica requisitos de interação e comunicação muito específicos, que têm que ser tidos em conta no desenvolvimento de uma interface.

1.3. Cnotinfor (Empresa proponente)

A Cnotinfor é uma empresa que desde 1988, se dedica à pesquisa e inovação tecnológica, na procura de soluções de aprendizagem e integração. Com sede em Coimbra, internacionalizou-se em 1993 para o Brasil e está desde 2001 também presente no Reino Unido. É constituída por pessoas de diversas áreas, desde engenheiros de reabilitação a engenheiros informáticos.

Ao longo destes anos criou inúmeras aplicações lúdicas para todos os graus de ensino, desenvolveu soluções de comunicação para pessoas com necessidades especiais e, formou professores e outros formadores, no uso das novas tecnologias para melhorar a aprendizagem e valorizar o ensino.

Assinando com a marca Imagina®, tem-se adaptado às necessidades e às novas tendências e inovações tecnológicas, nomeadamente à adaptação do seu software a dispositivos móveis (iOS, Android, Windows Phone, etc.).

1.4. Estágio

Para concretizar este projeto, a Cnotinfor realizou esta proposta de estágio apresentada ao Departamento de Engenharia Informática da Universidade de Coimbra (DEI), que foi então executado por mim, no âmbito da disciplina de Estágio/Dissertação, do Mestrado em Engenharia Informática.

O estágio dividiu-se em 2 semestres, tendo sido o 1º semestre dedicado principalmente ao estudo da envolvência do tema e do *Estado da Arte* da área tecnológica, ao levantamento de requisitos e especificação e, ao estudo das tecnologias escolhidas para a implementação do projeto, enquanto o 2º semestre foi mais focado na implementação e testes, apesar de a planificação do estágio ter sido alterada em virtude de alterações de especificação, como será mostrado mais à frente.

São orientadores de estágio, o Prof. Dr. Luís Macedo pelo DEI, e Eng. Nuno Machado pela Cnotinfor.

1.5. Equipa de trabalho

O projeto aqui apresentado é da completa exclusividade da Cnotinfor e da sua equipa de trabalho. Envolvidos ativamente no ActiveBrain estão Margarida Silva (engenheira de reabilitação e responsável pelo projeto) e o estagiário David Carvalho (gestor do projeto tecnológico). As decisões tecnológicas são da responsabilidade do estagiário, com a orientação de Nuno Machado (orientador do estágio e chefe da equipa de desenvolvimento), baseadas nos requisitos levantados e nas possibilidades da empresa. Já as decisões de especificação, design e outras não tecnológicas, são da responsabilidade da responsável pelo projeto, mas também Secundino Correia, sócio-gerente da empresa, que acompanha os projetos desenvolvidos. Por algumas decisões não caberem exclusivamente ao estagiário, nomeadamente a especificação, será normal ao longo deste relatório, por vezes ser mencionado a 1ª pessoa do plural.

1.6. Estrutura do documento

Este documento relata o desenvolvimento do estágio, desde a análise geral das necessidades de concretização do projeto ActiveBrain, passando pelo levantamento de requisitos e estudo das tecnologias até à sua implementação.

Depois de expostas as razões da existência deste projeto, apresentamos no capítulo 2 o *Estado da Arte* da estimulação cognitiva e social, desde os métodos vulgarmente usados na generalidade das instituições de acolhimento à terceira idade, até às aplicações vocacionadas para o estímulo cognitivo e tecnologias passíveis de serem usadas no desenvolvimento das mesmas.

No capítulo 3 especifica-se toda a aplicação a desenvolver, incluindo a caracterização do utilizador e o levantamento de requisitos funcionais e não funcionais.

Segue-se o capítulo 4 onde é exposta a arquitetura da plataforma, incluindo a aplicação e o jogo, a API de comunicação e as estruturas de dados.

O desenvolvido do projeto ao longo do ano é relatado no capítulo 5, onde são apresentados o planeamento relativo ao 1º semestre e ao 2º semestre, resumindo mais tarde qual o estado de conclusão das várias partes deste projeto.

Por último, no capítulo 6, temos as conclusões onde é resumido o estado do projeto, feita uma retrospectiva do estágio e uma autoavaliação do estagiário.

Capítulo 2

Estado da Arte

Sendo o ActiveBrain a implementação de uma solução para o envelhecimento ativo e para o combate à senescência e senilidade, importa analisar como é feito esse combate atualmente. Perceber quais os métodos e qual o alcance das práticas atuais, faz-nos entender melhor o nosso público-alvo, quais as falhas atuais, quais as mais-valias a replicar e as preocupações a ter no desenvolvimento desta plataforma. Importa também analisar a absorção da tecnologia por esta faixa etária e as suas possibilidades socioeconómicas de acesso à mesma.

Começemos por isso por analisar este público-alvo, seguindo-se uma análise as práticas tradicionais mantidas pela maioria das instituições de acolhimento do idoso, pela família ou por motivação do próprio, e depois, analisamos as soluções tecnológicas já existentes que atuam nesta área.

2.1. Estudo do público-alvo

No arranque deste projeto, ainda antes de se iniciar o estágio aqui relatado, foi feito um estudo^[11] por Margarida Lopes Silva, atual membro da equipa da Cnotinfor, no âmbito da dissertação final do seu Mestrado em Engenharia de Reabilitação e Acessibilidades Humanas, pela Universidade de Trás-os-Montes e Alto Douro.

Este estudo incluiu um inquérito dirigido a profissionais de reabilitação, terapeutas, geriatras, enfermeiros e cuidadores do idoso, pois na maioria dos casos os planos de reabilitação são efetuados por estes e porque nem sempre é possível obter a partir dos próprios idosos o que esperam de uma plataforma deste tipo.

Foram analisados quais os problemas cognitivos associados ao envelhecimento, para perceber quais as áreas de atuação a implementar no treino e, quais os indícios de demência e de que forma é que o declínio cognitivo está associado ao risco de demência. Percebeu-se que os profissionais de reabilitação consideram diversas áreas de cognição com diferentes graus de importância, nomeadamente do mais importante para o menos importante, as áreas da Memória, Funções Executivas, Concentração, Atenção, Velocidade de Raciocínio, Capacidade Linguística, Memória Auditiva e Visão Espacial. Neste estudo conclui-se também que a prática de jogos, a resolução de exercícios de lógica e a busca por uma vida social ativa, são considerados por todos, métodos mais eficazes para o combate ao declínio cognitivo.

No que toca a dispositivos, é bastante expressiva a preferência pelo Computador Pessoal, seguindo-se os tablets, apesar de as plataformas de treino serem ainda desconhecidas pela maioria dos profissionais da área.

2.2. Combate à senescência e senilidade nos dias de hoje

Atualmente existe uma maior preocupação no que toca ao envelhecimento ativo, e cada vez mais as entidades governamentais e não-governamentais dedicadas aos idosos, procuram manter programas que entretenham, estimulem e promovam a socialização das pessoas na terceira idade.

Estes programas incluem excursões a pontos de interesse cultural, jogos tradicionais, animação musical ou teatral no próprio lar, centro de dia ou outra valência onde habitualmente se encontram os idosos, etc.

Em suas casas, para os idosos que ficam ao cuidado das suas famílias ou muitos infelizmente, sozinhos, torna-se mais difícil promover o bem-estar mental, a atividade cerebral e a socialização. Existem no entanto instituições que promovem as chamadas Universidades Sênior, onde inúmeras atividades são praticadas, desde desporto, formação em diversas áreas como tecelagem, pintura, música, aulas de dança e poesia.^{[23][24]}

É importante que a família tenha noção destas necessidades sociais e lúdicas, no combate à senescência e senilidade, pois muitas vezes é com o incentivo e apoio dos seus filhos e restantes familiares, que o idoso integra planos de envelhecimento ativo, fora do seu meio residencial.

2.3. Plataformas “Ginásio Cerebral”

Em casa, centro social ou outra estrutura de acolhimento do idoso, é possível manter a mente do idoso entretida e estimulando as suas capacidades cognitivas. Se antigamente eram os jogos tradicionais que cumpriam esse papel, como o dominó, o xadrez ou as cartas, hoje, sem substituir os tradicionais jogos, é possível utilizar a tecnologia das aplicações para estimular de modo mais incisivo as capacidades mentais do ser humano. Desta forma, surgiram já algumas plataformas de treino cognitivo através de variados jogos, disponíveis como aplicação *stand-alone* ou através de portal web.

O estudo referido no ponto 2.1, incluiu um levantamento de plataformas de jogos cognitivos já existentes e estudadas a clareza e intuitividade das mesmas, requisito fulcral para um Público-Alvo não tão imergido na tecnologia como as faixas etárias mais novas. Com o início do meu estágio, alarguei o estudo à análise de mais funcionalidades e inclusive a quais incluem aplicações móveis e uso *offline*. Esta preocupação tem em conta a tendência crescente de utilização de aplicações em dispositivos móveis, como será referido mais adiante, mas também a possibilidade de o utilizador poder treinar o seu cérebro em qualquer lado mesmo sem ligação à Internet. Foram analisadas também, outras funcionalidades a implementar, que fossem mais valias para a este projeto.

De seguida é apresentada a comparação à análise realizada entre 3 plataformas. No primeiro semestre deste estágio foram testadas mais plataformas, mas não serão referidas aqui por não acrescentarem nada de diferente às mencionadas. Refiro apenas as melhores do ponto de vista de número e tipo de jogos, interoperabilidade, etc., para conseguirmos perceber em que falham, e o que é necessário colmatar com este projeto.

2.3.1. Lumosity (lumosity.com)^[3]

É uma plataforma *online* bastante completa, que inclui também 5 eixos de progresso cognitivo, um diagnóstico cognitivo inicial para configurar a dificuldade e as capacidades mais carentes de treino, e inclui uma variedade significativa de jogos, focados nas áreas da Memória, Velocidade de Raciocínio, Atenção, Flexibilidade, Funções Executivas, Concentração, Capacidade Linguística e Visão Espacial. Tem um registo de treino, que permite ao utilizador perceber como tem sido a sua evolução desde o início de uso da plataforma, quais as áreas de treino mais carenciadas e um histórico de que dias treinou. No entanto falha por não ter componente de socialização e por não possuir uma aplicação *offline*. Não tem componente de interação social e a plataforma é exclusivamente em inglês.

2.3.2. Fit Brains Trainer (fitbrains.com) ^[4]

Plataforma *online*, mas que está também disponível como aplicação móvel para Android e para iOS permitindo jogar *offline*. Tem treino personalizado, mas não permite vários níveis de dificuldade. Com jogos bastante variados, foca-se na Memória, Atenção, Funções Executivas, Concentração, Capacidade Linguística e Visão Espacial. Tem histórico de evolução, comparativo com a média da comunidade. Permite visualizar a pontuação dos amigos do Facebook que usam igualmente o FitBrains, no entanto não permite interagir com eles. É exclusivamente em inglês.

2.3.3. Cérebro Melhor (cerebromelhor.com.br) ^[5]

Sem aplicação *offline*, esta plataforma tem uma interface pouco limpa e amigável. Ganha (e foi incluída neste comparativo) por ser em português, no entanto fica muito atrás da maioria existentes. Tem no entanto uma boa variedade de jogos e tem também um diagnóstico inicial e treino adaptado. As áreas de atuação focadas são a Memória, Atenção, Funções Executivas e Visão Espacial.

2.3.4. Cognifit (cognifit.com) ^[6]

A Cognifit é uma plataforma muito completa no que toca a jogos, incluindo uma avaliação inicial de capacidades e treinos orientados às necessidades de cada jogador. Atua nas áreas da Memória, Velocidade, Atenção, Funções Executivas, Concentração, Capacidade Linguística, Visão Espacial e Memória Auditiva. Tem aplicação para uso *offline*, apesar de estar apenas disponível para iOS. Falha também por não ter nenhum estímulo social.

2.3.5. Comparativo

	Lumosity	Fit Brains Trainer	Cérebro Melhor	Cognifit
Diagnóstico Inicial	Sim	Sim	Sim	Sim
Treino adaptado às capacidades	Sim	Sim	Sim	Sim
Multi-plataforma	Website / iOS / Android	Website / iOS / Android	Website	Website / iOS
Interação Social	Não	Não	Não	Desafios assíncronos
Português	Não	Não	Sim	Sim

Áreas de atuação	Memória	×	×	×	×
	Velocidade	×			×
	Atenção	×	×	×	×
	Flexibilidade	×			
	Funções executivas	×	×	×	×
	Concentração	×	×		×
	Capacidade Linguística	×	×		×
	Visão Espacial	×	×	×	×
Memória Auditiva				×	

Tabela 1 - Comparativo entre aplicações de treino cerebral existentes

Verificamos que as plataformas Fit Brains e Lumosity destacam-se por incluírem uma aplicação para smartphones e tablets de vários sistemas operativos. No entanto não existe em língua portuguesa, e não tem qualquer tipo de interação social. A plataforma Cognifit destaca-se por incluir já alguma interação social, não passando no entanto de desafios assíncronos, ie., um jogador pode desafiar outro a jogar o mesmo jogo para que no fim haja comparação da pontuação obtida.

2.2. Conclusão

Os programas de Universidades Sénior, Centros de Dia e outras instituições devidamente formadas nesta temática do envelhecimento ativo, são formas de estimulação mental e social, dificilmente ultrapassadas por qualquer outro método. No entanto, e tendo em conta o isolamento de muitos idosos, a própria vontade de estar em casa ou outros motivos que os afastem dessas instituições, uma plataforma de treino acessível através de meios tecnológicos ao alcance de (quase) qualquer um, nomeadamente usando computador, tablet ou smartphone, ultrapassaria assim o problema da falta de acompanhamento pessoal por especialistas da área da psicogeriatría.

Analisando as 4 plataformas (e outras que não foram aqui incluídas) e percebendo que algumas estão já bastante desenvolvidas, destaca-se no entanto que nenhuma tem um dos 2 componentes chave que serão uma mais valia no nosso projeto: Socialização ativa e acompanhamento profissional.

Capítulo 3

Especificação

Após a análise do Estado da Arte, ficámos a perceber as necessidades do nosso Público-Alvo e as capacidades das soluções já existentes. Partimos assim em conjunto, para a especificação dos requisitos funcionais. Estes incluem um levantamento de requisitos de layout e de funcionalidades a implementar, tendo em conta os objetivos pretendidos e a especificidade do público-alvo em questão, que sendo maioritariamente idoso, requer algum cuidado na complexidade das interfaces, confirmação de ações, etc.

3.1. Descrição Geral

O ActiveBrain é uma plataforma que visa o envelhecimento ativo, combatendo a senilidade, o isolamento, aproximando os mais idosos da sua família e da sociedade em geral, e facilitando o diagnóstico médico no que toca às suas funções cognitivas.

Com base neste conceito existem 3 componentes chave nesta plataforma, nomeadamente Jogos, Partilha de histórias e Divulgação de notícias. De seguida descreve-se mais ao pormenor cada uma, bem como a plataforma que as irá integrar:

Jogos

A mais importante componente desta plataforma, pois é a que consegue um estímulo cognitivo mais eficaz e que pode ser adaptado de pessoa para pessoa, consoante as suas necessidades e capacidades. Pretende-se que existam uma variedade de jogos, em que cada um trabalhe áreas específicas de cognição: Memória, Velocidade, Flexibilidade, Resolução de Problemas e Atenção. Os jogos terão que ser simples e intuitivos, de modo a não cansar o utilizador e o motivar a completá-lo. Cada jogo poderá ter diferentes níveis de dificuldade, consoante as capacidades do utilizador e igualmente resultados (pontos) com ponderação do estado cognitivo de cada um.

Importante também no jogo é o fator socialização, que irá estimular ainda mais o idoso a se exercitar mentalmente e a comunicar com outros. Esta socialização será feita por partilha de jogo (jogarem 2 pessoas em equipa) ou competição (um contra o outro), através de constante comunicação por chat e ainda pela possibilidade de todos os utilizadores partilharem a sua opinião sobre o jogo, num painel de comentários e pontuação. Este fator socialização, irá diferenciar o ActiveBrain de outras plataformas semelhantes.

Partilha de histórias

Nem sempre o idoso tem com quem falar, com quem partilha as suas memórias, as suas ideias. Este fato inibe a estimulação mental e acelera a senilidade. Queremos portanto com o ActiveBrain, dar a possibilidade de o idoso partilhar as suas histórias com outros, dizer o que lhe vai na alma e criar assim mais um canal de comunicação entre os vários utilizadores. As histórias serão públicas, mas terá que haver destaque para as histórias dos amigos de cada um.

Divulgação de notícias

O idoso nem sempre está em contato com o mundo, nem sempre tem assuntos que lhe proporcionem uma conversa. A divulgação de notícias pode assim, ser um estímulo para a

lembrança de histórias, para uma conversa, para recorrer à sua memória. Estas serão no entanto notícias selecionadas com esta finalidade, através da própria Cnotinfor.

Plataforma

Acima foram descritas resumidamente as 3 componentes chave do ActiveBrain. É no entanto necessária uma plataforma que as integre, e que faça a gestão de utilizadores, de conexão, etc.

O ActiveBrain inclui um sistema de *login* e registo que permite ao utilizador iniciar-se na plataforma com o seu perfil, com as suas definições, seus amigos, etc. Igualmente existe uma área de definições pessoais, onde alguns dados podem ser alterados, como por ex. a password de acesso. Além disso, cada utilizador pode definir uma lista de amigos, com os quais poderá jogar e conversar, e cujas suas histórias e interação terão destaque em relação aos demais utilizadores.

3.2. Caracterização do utilizador

Para a caracterização do utilizador, perceber as necessidades, limitações e expectativas do mesmo, foi de extrema importância o estudo efetuado por Margarida Silva, já referido no capítulo sobre o Estado da Arte deste relatório. É no entanto necessário também, perceber as limitações físicas e mentais do utilizador, para conseguir adequar a aplicação a um utilizador de terceira idade.

A faixa etária do nosso público-alvo exige especificações muito concretas no que toca à interface de utilizador. É de sabedoria comum que o envelhecimento traz reduções na capacidade de perceção e na velocidade de interação. Há no entanto mais aspectos a ter em conta, que irão afetar não só o design, mas também a resposta da interface, a arquitetura de comunicação, etc. Falando de um público que por vezes rejeita a tecnologia e receia o seu uso^[17], é importante criar mecanismos de interação fáceis, intuitivos e agradáveis, para que este se sinta confortável nas suas ações e consiga acompanhar e ter a certeza do que está a fazer.

Com o envelhecer surgem vários problemas físicos que afetam a nossa capacidade^[20] de perceção, de visualização, etc. O enfraquecimento da nossa habilidade visual é uma delas. Com a idade, mais concretamente a partir dos 60 anos, o campo de visão diminui e o contraste de cores torna-se menos nítido. Em alguns casos há mesmo confusão e pouca/nenhuma distinção entre cores, nomeadamente no campo azul-verde.

A quantidade de informação e o ruído visual são também um aspeto a ter em conta neste público. O idoso distrai-se mais facilmente com formas ou imagens desnecessárias, podendo assim perder o seguimento à operação que estava a realizar. É aconselhável portanto, aquando da criação de interfaces para idosos, o uso apenas das imagens e textos necessários e relevantes para a execução uma tarefa ou simplesmente para comunicar com o utilizador.

Importante também é obrigar o utilizador a memorizar o menos possível (excepto claro, em jogos cuja intenção seja precisamente estimular o campo da memória). Isto implica que seja a aplicação a memorizar possíveis dados de passos anteriores, a facilitar o sistema de *login*, etc.

Para concluir a descrição do utilizador, refira-se que com a idade a velocidade motora diminui e portanto, é necessário uma tolerância maior a ações que tenham em conta intervalos de tempo entre 2 interações (por ex. duplos-cliques) ou à leitura de informação antes de alternar uma cena automaticamente. No desenvolvimento de interfaces de *touch*, não nos podemos também esquecer que a precisão de uma mão idosa é bastante diferente do habitual, havendo mesmo casos mais complexos, como é o caso dos idosos com a *Doença de Parkinson*.

3.3. Dispositivos

O desenvolvimento de uma aplicação como esta é bastante longo, e por isso é necessário ter em conta que decidimos certamente qual o dispositivo alvo da aplicação, pois é diferente a implementação das mesmas funcionalidades em Windows, Android, etc.

Conforme estudo já referido^[11], efetuado por Margarida Silva, a tendência dos profissionais de geriatria e reabilitação é preferirem o computador para o uso de aplicações deste género. No entanto a normal evolução da tecnologia tem acentuado o uso de tablets e o mercado dos tablets é hoje, mais acessível e mais completo. O *tablet* tem por si só a excelente vantagem da sua portabilidade, o que permite por exemplo, que um idoso tanto pratique a sua mente em casa, como em casa dos seus familiares, no centro e dia, etc. Além disso, segundo o estudo de Tanid Phiriyapokanon^[17], o uso de tablets é mais eficaz junto de utilizadores mais idosos.

Os *tablets* no entanto, à semelhança de um *Desktop* ou *Laptop*, podem albergar diferentes sistemas operativos, e consequentemente diferentes implementações da mesma aplicação. No mercado dos tablets é o sistema Android que domina^[20] seguido do sistema operativo da Apple (iOS) e por fim, Windows.

Se conseguirmos no entanto, desenvolver um software em que seja possível a migração fácil para diversos dispositivos e sistemas operativos, então não necessitaremos de optar por um mercado específico, e conseguiremos levar o ActiveBrain a um maior número de utilizadores que irão assim beneficiar desta plataforma.

3.4. Análise de Requisitos

A análise de requisitos reflete o que se espera do comportamento do software, tanto a nível de interação com o utilizador (requisitos funcionais) como também as necessidades “não-visíveis”, como por exemplo especificações que garantam segurança, confidencialidade, persistência, etc.

3.4.1. Requisitos Funcionais

Os requisitos listados a seguir, e mais detalhados no Anexo A, foram atualizado conforme o novo conceito do ActiveBrain discutido e decidido no decorrer do 2º semestre. As funcionalidades enumeradas encontram-se agrupadas por componentes (já descritas anteriormente), juntamente com o grupo de funcionalidades de autenticação e registo, menu principal e por fim, as funcionalidades referentes ao jogo específico, parte deste projeto de estágio.

O projeto desenvolvido inclui uma plataforma com uma área de edição de dados pessoais, gestão de amigos e painel informativo do progresso, um gestor de jogos com ordenação por categoria e/ou por área cognitiva a desenvolver, consoante as necessidades do utilizador, um painel de visionamento de notícias de interesse e uma área de visionamento e publicação de histórias. Por fim, nesta fase inicial do projeto, foi incluído no planeamento, o desenvolvimento de 1 jogo completo. A plataforma está no entanto concebida para albergar diversos jogos, pelo que existem funcionalidades comuns a todos os jogos, que foram já implementadas.

De seguida apresentam-se os requisitos funcionais da aplicação, aqui resumidos para uma listagem melhor perceptível. Para mais detalhes sobre os mesmos por favor consultar o Anexo A.

Painel de Autenticação

[RF101]	Login com e-mail	Tipo: Muito Importante
[RF102]	Login automático	Tipo: Importante
[RF111]	Registrar-se	Tipo: Importante
[RF103]	Login selecionando utilizador	Tipo: Neutro
[RF112]	Experimentar gratuitamente	Tipo: Pouco importante
[RF121]	Sair	Tipo: Pouco importante

Menu Principal

[RF201]	Escolha de funcionalidade (atualizado no 2º semestre)	Tipo: Muito importante
[RF202]	Notificação de mensagens	Tipo: Nada importante
[RF203]	Logout	Tipo: Nada importante

Perfil de Utilizador

[RF331]	Ver diagnóstico	Tipo: Muito Importante
[RF341]	Voltar ao Menu Principal	Tipo: Muito Importante
[RF301]	Edição de perfil	Tipo: Importante
[RF311]	Ver amigos	Tipo: Importante
[RF312]	Pesquisar pessoas	Tipo: Importante
[RF302]	Edição de foto	Tipo: Pouco Importante
[RF313]	Convite a amigos	Tipo: Nada Importante
[RF321]	Ver mensagens	Tipo: Nada Importante

Painel de Jogos

[RF401]	Listar jogos	Tipo: Muito Importante
[RF404]	Voltar ao Menu Principal	Tipo: Muito Importante
[RF402]	Informação sobre o jogo	Tipo: Importante
[RF403]	Modo de jogo	Tipo: Importante

Ver histórias

[RF531]	Voltar ao Menu Principal	Tipo: Muito Importante
[RF501]	Listar histórias (atualizado no 2º semestre)	Tipo: Importante
[RF511]	Escrever história (atualizado no 2º semestre)	Tipo: Importante
[RF521]	Eliminar história (atualizado no 2º semestre)	Tipo: Importante

Ler notícias

[RF611]	Voltar ao Menu Principal	Tipo: Muito Importante
[RF601]	Listar notícias (atualizado no 2º semestre)	Tipo: Importante

Modo de jogo

[RF711]	Janela de chat (para jogos multijogador)	Tipo: Muito Importante
---------	--	-------------------------------

[RF712]	Caixa para envio de mensagem de chat (para jogos multijogador)	Tipo: Muito Importante
[RF721]	Painel de resumo final em jogo singular	Tipo: Muito Importante
[RF731]	Painel de resumo final singular em jogo multijogador	Tipo: Muito Importante
[RF781]	Sair do jogo	Tipo: Muito Importante
[RF701]	Instruções iniciais	Tipo: Importante
[RF741]	Painel de resumo final - Comparativo com outros	Tipo: Importante
[RF761]	Feedback	Tipo: Importante
[RF771]	Jogo seguinte	Tipo: Importante
[RF751]	Sugerir jogo a amigo	Tipo: Nada Importante
Jogo específico		
[RF801]	Mapa inicial	Tipo: Muito Importante
[RF811]	Barra de tempo inicial	Tipo: Muito Importante
[RF821]	Remover objetos	Tipo: Muito Importante
[RF831]	Barra de tempo final	Tipo: Muito Importante
[RF841]	Traçamento de caminho	Tipo: Muito Importante
[RF851]	Alcance do ponto final	Tipo: Muito Importante
[RF861]	Terminus do tempo	Tipo: Muito Importante

Tabela 2 - Lista de requisitos funcionais da aplicação

3.4.2. Requisitos não-funcionais

3.4.2.1. Persistência de dados

O ActiveBrain irá gerar dados mensuráveis a cada jogo, não só a pontuação mas também as capacidades demonstradas no que toca aos 5 eixos de progresso cognitivos, Memória, Atenção, Flexibilidade, Resolução de Problemas e Velocidade. Sendo o ActiveBrain uma plataforma de treino cerebral que pretende fazer um acompanhamento e gerar progresso nas capacidades cognitivas do utilizador, os dados relativos a estes fatores terão que ser persistentes. A especificar também, está a necessidade de não limitar os dados de um determinado utilizador a um dispositivo apenas. Pretende-se que o ActiveBrain funcione por conta de utilizador, e não por dispositivo, permitindo assim que um utilizador possa tanto utilizá-lo no seu *Desktop/Laptop* como no seu *tablet*, ou em muitos casos, em *tablets* ou *Desktops* públicos que nem sempre são os mesmos.

Os dados gerados serão guardados em duas bases de dados distintas: Armazenamento Local e Servidor Central. O dados a guardar e seu tipo, serão mais à frente descritos na Arquitetura da Aplicação.

Armazenamento Local

Nem sempre a Internet poderá estar acessível, e de forma a não limitar o uso da aplicação apenas a quando houver conexão à rede, o ActiveBrain terá que ser capaz de ser usado no máximo das suas funcionalidades possíveis, mesmo estando *offline*. Para isso, irá guardar

localmente dados que permitam funções básicas como login, consulta do progresso cognitivo, consulta de mensagens recebidas e claro, jogar

Armazenamento Remoto

Havendo inúmeros utilizadores que interagem entre si, é necessário haver sincronização entre os diversos dispositivos em utilização. Para isso será criado um servidor de armazenamento de dados onde toda a informação será centralizada, e onde serão guardados todos os dados, independentemente de já estarem guardados nos dispositivos locais. Como não queremos limitar o uso do ActiveBrain a um só dispositivo, é necessário que ao utilizar outro dispositivo, este consiga iniciar-se preferencialmente com os dados da última sessão, saber quais os amigos do utilizador, etc.

3.4.2.2. Sincronização de dados

Havendo 2 locais de armazenamento (local e remoto), a aplicação deverá, sempre que forem feitas alterações locais, sincronizar com o armazenamento remoto caso haja conexão disponível. Deve no entanto evitar tráfego desnecessário com vista a reduzir possíveis custos associados ao tráfego em redes móveis. Para isso deverá, antes de transferir alguns dados como por exemplo fotografias de utilizadores, confirmar se estas foram alteradas remotamente.

Capítulo 4

Arquitetura do Sistema

4.1. Visão geral

Como foi descrito no capítulo anterior ao abordar os requisitos, o ActiveBrain inclui um servidor remoto onde o armazenamento central e sincronização de dados será feito. A aplicação a correr nos dispositivos, não comunica diretamente com o sistema de gestão de base de dados remotos (DBMS). É necessário uma interface intermédia que responda a pedidos HTTP provenientes da aplicação. Esta decisão deve-se a múltiplos fatores. Um deles, como será explicado no capítulo seguinte, é que a *framework* escolhida para programar a aplicação, não comunica diretamente com DBMS, pelo que necessita de comunicar através de pedidos HTTP. Outro motivo é que ao criar uma interface que receba pedidos HTTP e retorne XML, ficamos com uma API pronta a comunicar com qualquer tipo de *front-end*, seja uma aplicação móvel ou um website usando Javascript ou mesmo uma aplicação *server-side* como o PHP. Isto poderá ser útil por exemplo, aquando da implementação da plataforma de análise de dados cognitivos, para médicos, geriatras e outros profissionais de apoio ao idoso.

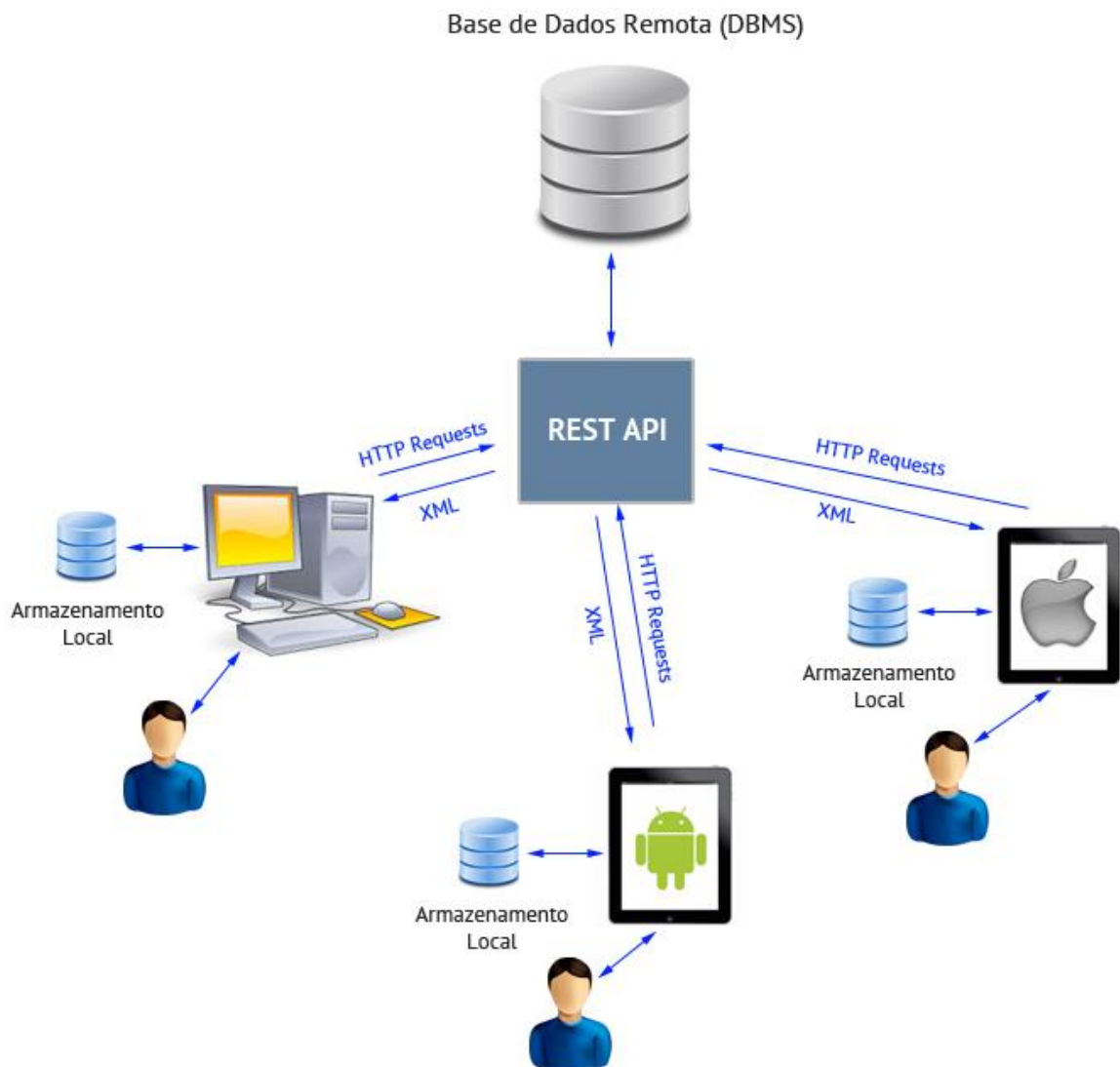


Figura 1 - Representação da Arquitetura do Sistema

4.2. Estrutura de dados

4.2.1. Estrutura de dados locais

Como já foi referido, pretende-se que mesmo *offline*, seja possível usar o máximo das funcionalidades possíveis do ActiveBrain. Deste modo, quase todas as funcionalidades estarão disponíveis pelo menos em modo leitura, tal como descritas na Análise de Requisitos Funcionais. Para isso, é necessária uma estrutura de dados quase tão complexa como a estrutura de dados remota. Será no entanto claro está, menos populada pois serão dados referentes apenas aos utilizadores do dispositivo em questão.

Segue-se o Diagrama Relacional que descreve os dados a manter localmente e seu tipo:

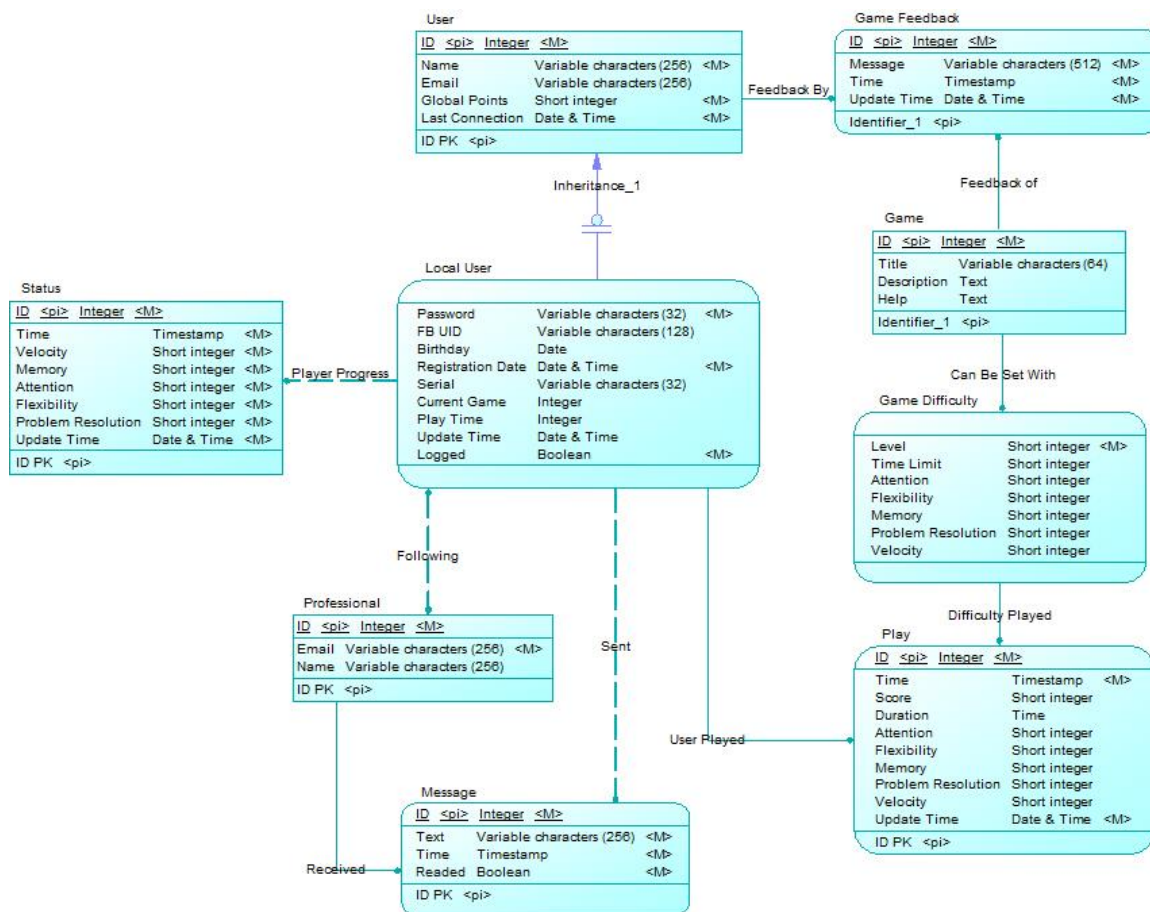


Figura 2 - Diagrama ER de dados locais, do ActiveBrain

4.2.2. Estrutura de dados remotos

Remotamente num servidor centralizado, serão guardado os dados de todos os utilizadores, pontuações, progresso cognitivo, *feedback* de jogo, etc.

Esta centralização é necessária para que a aplicação seja multi-dispositivo e um mesmo utilizador possa usar diferentes dispositivos em diferentes sessões. O repositório de dados remoto assegura ainda, que no caso de desinstalação ou perda de dados local, o utilizador não perde histórico de jogo, de interação e progresso cognitivo. Além disso, no caso de

desconexão inesperada, dados enviados para um determinado utilizador (mensagens, chat) nunca serão perdidos.

Segue-se o Diagrama Relacional de dados que a base de dados remota irá implementar:

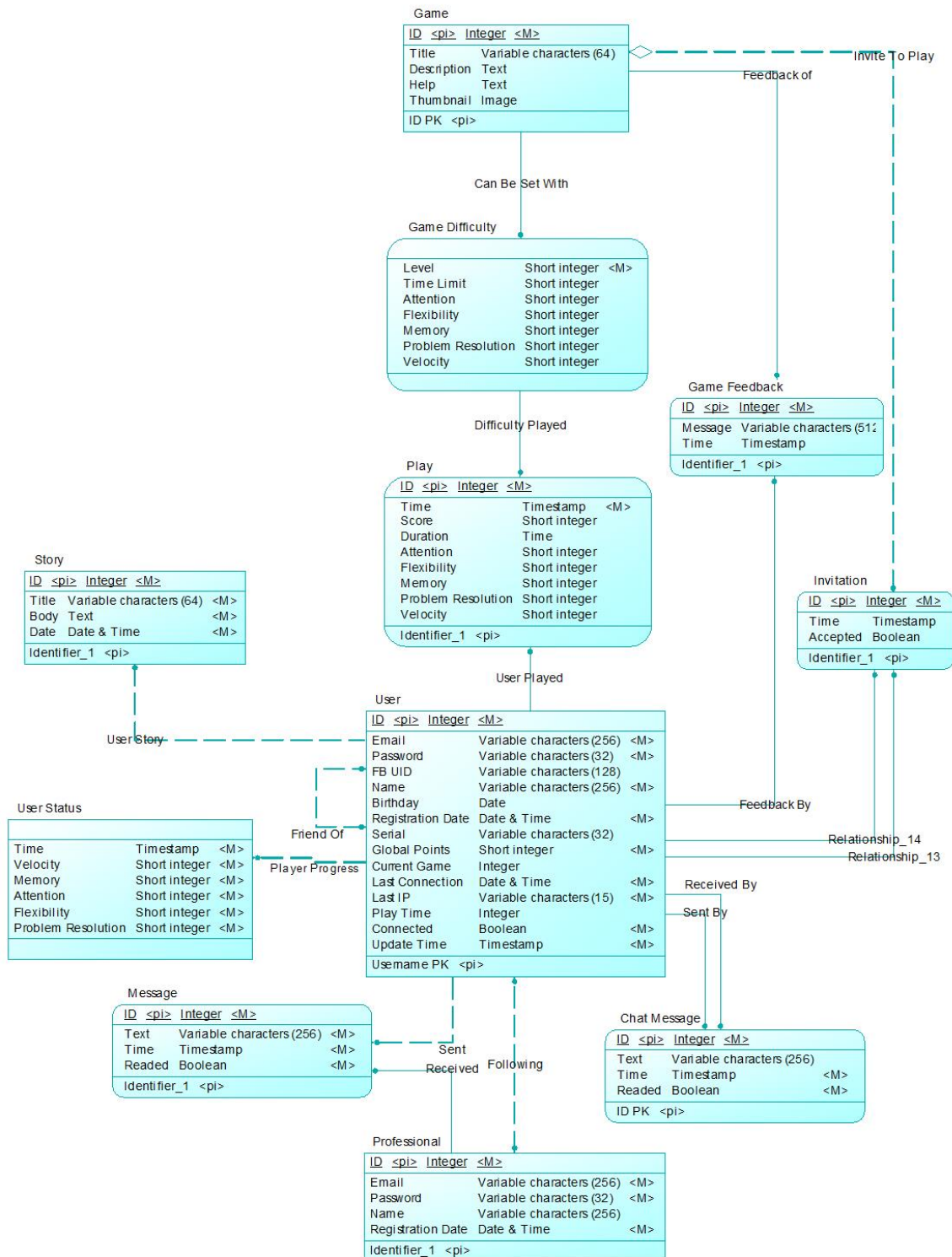


Figura 3 - Diagrama ER de dados remotos, do ActiveBrain

4.3. Estrutura de Classes

A implementação do ActiveBrain, segue a arquitetura MVC (Model-View-Controller), conseguindo-se deste modo, dividir o projeto em vários sub-projetos que quase se podem abstrair uns dos outros na sua implementação. Esta arquitetura permite grandes ganhos na eficácia e eficiência do desenvolvimento. Eficácia pois para efetuar pequenas alterações por exemplo a uma interface, não é necessário alterar todo o projeto de software. Eficiência pois esta arquitetura permite uma divisão fácil e clara de tarefas entre equipa, bem como maior velocidade de desenvolvimento e compilação.

Temos portanto classes para cada módulo desta arquitetura, como se pode ver na figura seguinte:

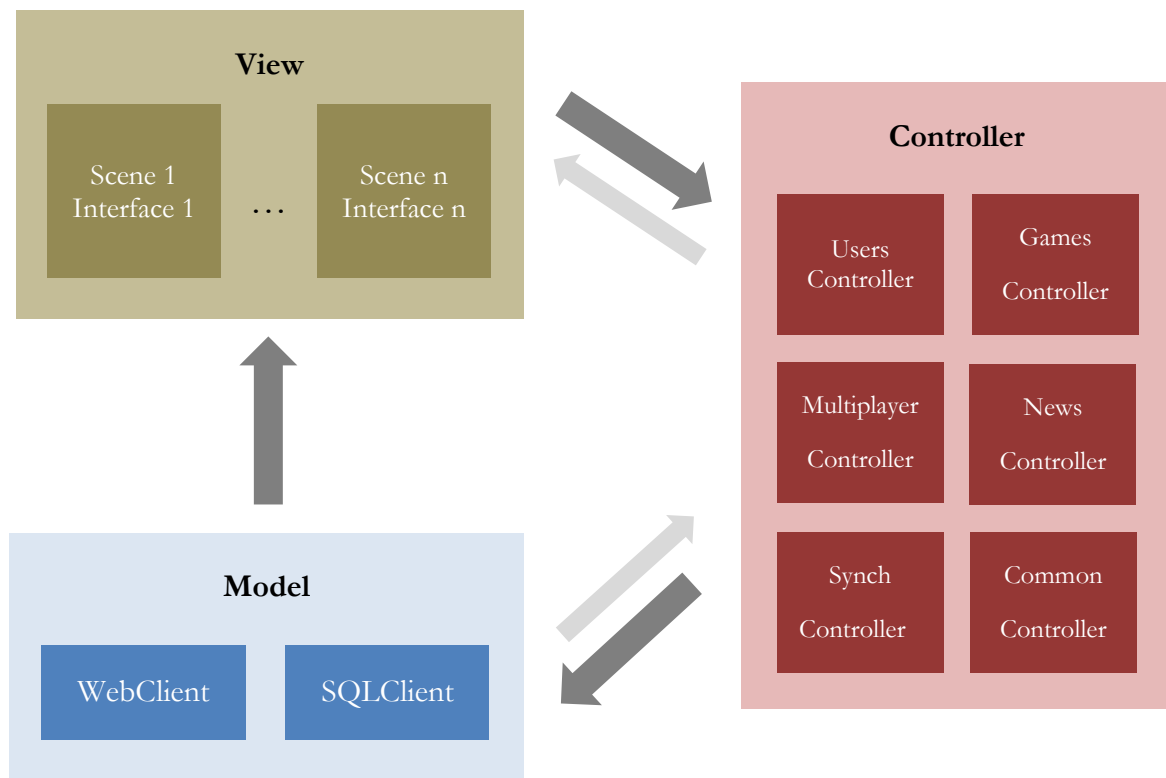


Figura 4 - Estrutura de classes implementada em MVC

Claro está, que esta não é uma representação de todas as classes usadas na aplicação, mas sim das mais relevantes em termos de arquitetura.

4.4. API REST

Para a comunicação com o servidor, o estagiário decidiu criar uma API REST, impulsionado pela restrição da tecnologia de desenvolvimento (mais à frente descrita), mas também porque o protocolo HTTP é o mais suportado e desenvolvido meio de comunicação com máquinas externas, por todas as linguagens de programação de alto nível. Este fato poderá ser vital mais tarde, por se pretender que futuramente haja também outras aplicações a aceder aos dados centralizados (por ex. um website onde médicos, terapeutas e familiares seguem o desenvolvimento cognitivo do seu paciente/familiar. Esta API irá ser responsável pela comunicação com a base de dados e irá devolver códigos HTTP e dados em XML, sempre respeitando o protocolo HTTP/1.1.^[21]

4.4.1. Segurança

Esta API não pode estar acessível a qualquer um, por questões de privacidade e confiabilidade, pelo que foi necessário criar um sistema de acesso restrito. Para tal é usado o método HTTP Basic Authentication (BA), que requer o envio de um *username* e *password* como cabeçalho de qualquer comunicação HTTP com a API. Caso contrário o cliente HTTP irá receber o código 401 (Forbidden).

4.4.2. Apache mod_rewrite e URIs

Visionando a estrutura de dados do ActiveBrain como a interação entre diferentes entidades e seus atributos, a API REST foi criada tentando ao máximo cumprir uma hierarquia de subordinação e atribuição. Para tal, as funcionalidades de transformação de URL que o mod_rewrite do Apache permitem, foram usadas para criar pontos de acesso a objetos específicos, incluindo sua modificação ou eliminação, através de URIs claros e lógicos.

Por exemplo, para aceder aos dados de um determinado utilizador, é feito um http GET usando o URI http://{servidor:porto}/activebrain/{user_id}/. Se a aplicação necessitar de aceder à sua lista de amigos, irá obtê-la através do URI “http://{servidor:porto}/activebrain/{user_id}/friends”.

Por detrás destes URIs, estão diversos scripts PHP para os quais o Apache mod_rewrite redireciona os pedidos HTTP e que estes sim, comunicam com a base de dados MySQL e transformam as suas respostas em XML.

Em anexo (Anexo A) a este documento, encontram-se alguns exemplos de chamadas à API REST, seus parâmetros e dados devolvidos.

4.4.3. Respostas HTTP e XML

O XML é usado como o padrão nas respostas do servidor, pois os dados requeridos têm uma estrutura a respeitar que necessita de uma representação. De entre as mais comuns formas de representar dados estruturados (XML e JSON) foi escolhido o XML pela sua fácil construção e por existirem inúmeras bibliotecas de leitura de XML em C++, incluindo a TinyXML2 que faz parte das extensões da Cocos2d-x.

Quando não é possível devolver dados em XML, seja por falha na construção do URI, por inexistência dos dados pedidos ou até por não alteração desde a última consulta, são devolvidos códigos HTTP, pois estes limitam-se ao envio do cabeçalho HTTP, poupando assim no tráfego e acelerando a velocidade de resposta. Por exemplo no caso de acesso a um URI que representa um utilizador não existente, a API irá retornar o código HTTP 404 (Not found). Quando uma aplicação tentar obter uma fotografia de utilizador, envia a *hash* da fotografia que de momento contém. Caso a fotografia ainda seja a mesma, a API irá devolver HTTP 304 (Not modified), poupando assim o tráfego que se iria gerar pela transferência de uma imagem.

4.4.4. Métodos HTTP

A API do ActiveBrain usa 4 dos métodos de HTTP *requests*, nomeadamente GET, PUT, POST e DELETE, sendo que, cumprindo as recomendações do protocolo HTTP/1.1, os métodos GET, PUT e DELETE são idem-potentes, i.e., vários *requests* iguais irão produzir o mesmo efeito que apenas um único *request*, enquanto que o método POST não é idem-potente.

4.5. Fluxo de Dados

Sendo o ActiveBrain uma plataforma que usa fontes de dados externa e/ou interna, consoante o dispositivo esteja *online* ou *offline*, a aplicação tem, como foi mostrado no Diagrama de Classes (4.3), duas classes Modelo (arquitetura MVC): SQLClient e WebClient. Estas são responsáveis respetivamente, pela leitura da base de dados local (SQLite) e pelo acesso aos dados remotos, fornecidos pela API REST.

Assim cada classe Controladora da arquitetura MVC, na maioria das suas chamadas, verifica primeiro se o dispositivo está conectado à Internet para seleccionar assim, qual o Modelo a usar.

Apresenta-se de seguida, um exemplo de fluxo de dados referente à validação das credenciais de utilizador:

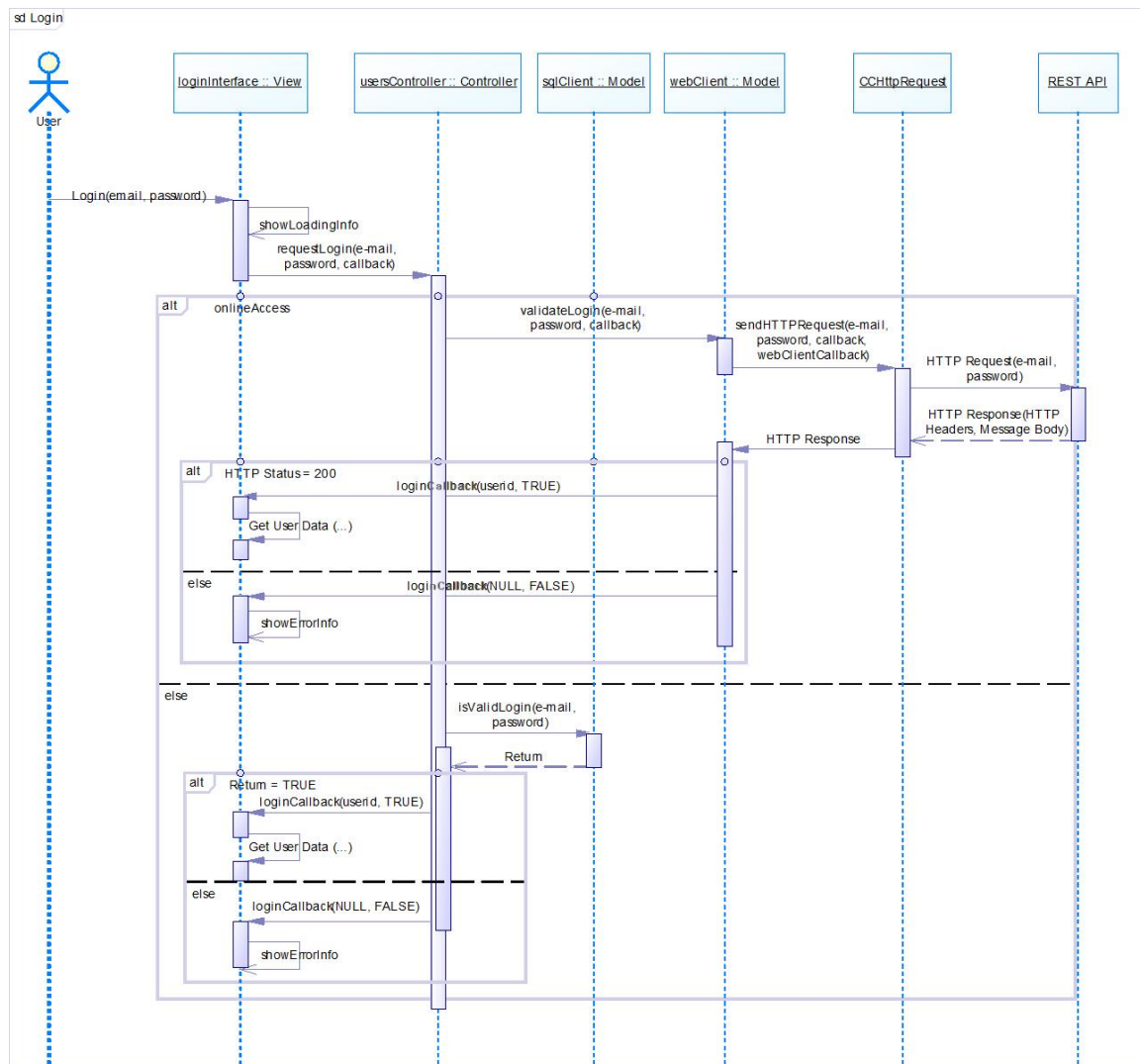


Figura 5 - Diagrama de Sequência da chamada de validação de Login

4.6. Lógica do jogo

Consistindo o jogo “Qual o melhor caminho” na escolha de um caminho entre muitos possíveis, cada um com seus objetos e respetiva pontuação, os seus mapas estão criados como um grafo orientado, no qual o utilizador terá que escolher um caminho entre uma origem definida e um destino, sem poder efetuar ciclos e respeitando a orientação das ligações entre os nós. O jogo contém vários níveis, cada um com diferentes mapas de modo a diminuir a probabilidade do utilizador escolher o melhor caminho por memorização do mapa completo, ao invés do esperado, que é o cálculo da pontuação através dos vários caminhos.

Os mapas são definidos através da definição de posicionamento dos nós do grafo orientado e dos valores para cada ligação.

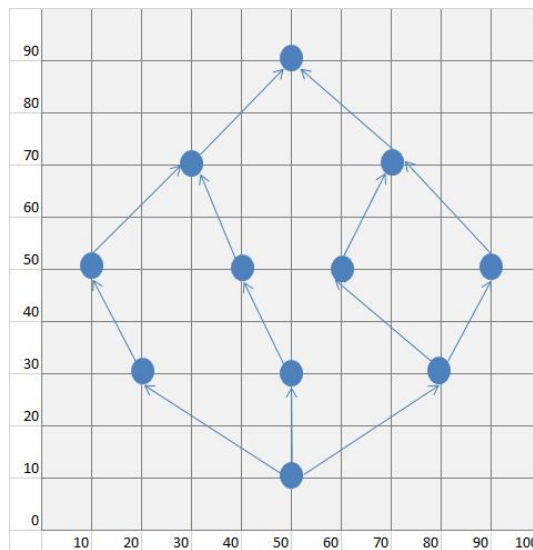


Figura 6 - Esboço de um mapa com o posicionamento dos pontos definido em percentagem

Para facilitar o carregamento dos mapas e posterior fácil atualização dos mesmos, os mapas são depois criados sob a forma de XML, por esta linguagem permitir a definição da hierarquia de níveis e mapas, bem como a definição de atributos:

```
<level id="3" time1="20" time2="10" graphs="2" attention="30" flexibility="15" memory="60" problem_resolution="80" speed="1"
  <graph id="1" max_points="40">
    <nodes>
      <node id="0" step="0" x="50" y="20"></node>
      <node id="1" step="1" x="30" y="40"></node>
      <node id="2" step="1" x="70" y="40"></node>
      <node id="3" step="2" x="20" y="60"></node>
      <node id="4" step="2" x="50" y="60"></node>
      <node id="5" step="2" x="80" y="60"></node>
      <node id="6" step="3" x="50" y="80"></node>
    </nodes>
    <edges>
      <edge from="0" to="1" points="10"></edge>
      <edge from="0" to="2" points="20"></edge>
      <edge from="1" to="3" points="-50"></edge>
      <edge from="1" to="4" points="10"></edge>
      <edge from="2" to="5" points="110"></edge>
      <edge from="3" to="6" points="60"></edge>
      <edge from="4" to="6" points="90"></edge>
      <edge from="5" to="6" points="-30"></edge>
    </edges>
  </graph>
  <graph id="2" max_points="40">
    <nodes>
      <node id="0" step="0" x="50" y="20"></node>
      <node id="1" step="1" x="30" y="40"></node>
      <node id="2" step="1" x="70" y="40"></node>
```

Figura 7 - Exemplo da definição de um mapa de jogo em XML, com sua hierarquia e atributos

Capítulo 5

Tecnologias

Este capítulo, que apresenta uma importante parte do trabalho de desenvolvimento de *software*, descreve as tecnologias escolhidas para o desenvolvimento deste projeto em todas as suas componentes. A escolha das tecnologias a usar foi também (além do desenvolvimento), um trabalho realizado pelo estagiário, que teve em conta várias soluções existentes no mercado e seus prós e contras, ao mesmo tempo que tentou responder às capacidades e facilidades existentes na empresa.

5.1. Framework / Linguagem de desenvolvimento da aplicação

No 1º semestre deste estágio foram analisadas várias linguagens de desenvolvimento e frameworks disponíveis no mercado, que conseguissem implementar aplicações multi-plataforma, como se pretende com o ActiveBrain.

De seguida mostra-se um comparativo de várias linguagens/frameworks analisadas e seus prós e contras, para mais à frente justificar a escolha feita não só por esta análise mas com influência de outros fatores. Mais opções existem, não estando no entanto listadas por não acrescentarem nada de novo ou relevante às que aqui serão apresentadas para comparação.

5.1.1. Análise de linguagens/frameworks existentes

HTML5

O uso das novas funcionalidades do HTML e de Javascript, tem vindo a permitir que o HTML5 esteja a receber imensas aplicações *online*, que antigamente não poderiam correr num browser. O HTML5 tem a vantagem de ser suportado por todos os *browsers* mais recentes, sejam eles de dispositivos móveis, como smartphones e tablets ou de dispositivos fixos (PC, Mac, etc.). No entanto, carece da instalação prévia do referido *browser* e, não conseguiríamos ter a aplicação a correr *offline*, sem por em risco a integridade e segurança do nosso código e dos dados gerados pela aplicação, pois o código HTML5 e Javascript poderiam ser acedidos.

Unity ^[7]

Unity é uma ferramenta de desenvolvimento de jogos 2D e 3D, com auxílio de um ambiente gráfico próprio onde se pode editar objetos, movimentos e cenários, bem como programar em C#, Javascript e Boo. Exporta aplicações para iOS, Android, Windows, etc., e ainda para consolas, como é o caso da Wii, PS3, e Xbox. Tem uma renderização com imensa qualidade, ideal para jogos 3D. É no entanto uma ferramenta complexa em que se perde imenso tempo só para a configurar para a primeira utilização. Além disso, tem alguns funcionalidades pagas, como por ex. a reprodução de vídeos.

Cocos2d-x ^[8]

O Cocos2d-x é uma Framework que disponibiliza ao programador um desenvolvimento na linguagem C++, Javascript e Lua e, está preparada para a conversão de uma aplicação feita em Cocos2d-x, em projetos de iOS, Android, Windows, Blackberry etc., abstraindo o programador não só da linguagem e da estrutura de ficheiros de cada dispositivo, mas também das características da máquina onde será a aplicação executada.

Suporta IDEs variados, visto ser um conjunto de bibliotecas de conhecidas linguagens, pelo que facilmente um programador se familiariza com o seu desenvolvimento no que toca à interface e configurações. É distribuída gratuitamente, com scripts para a criação de projetos multiplataforma, pelo que o programador não necessita de conhecer a estrutura de ficheiros e dados de todos os dispositivos.

Marmalade ^[9]

Marmalade é uma Framework semelhante ao Cocos2d-x, também desenvolvida em C++ e com a possibilidade de ser desenvolvida em inúmeros IDEs. Tem no entanto algumas funcionalidades pagas, como é o caso da maioria das extensões nativas.

5.1.2. Comparativo

	Multipluralidade de IDE's	Segurança	Gratuidade	Complexidade	
HTML5	+	-	+	+	Legenda: + Apreciação positiva - Apreciação negativa +/- Apreciação neutra
Unity	-	+	+/-	-	
Cocos2d-x	+	+	+	+/-	
Marmalade	+	+	+/-	+/-	

Tabela 3 - Comparativo entre frameworks de implementação multidispositivo

5.1.3. A escolha da framework Cocos2d-x

No comparativo anterior, verificámos que entre as bibliotecas/frameworks analisadas, destacam-se o HTML5 e o Cocos2d-x. O HTML5 apesar de positivo na maioria dos fatores analisados e ainda, da vasta documentação *online* sobre esta linguagem, falha no entanto pela falta de segurança, por ser uma linguagem não compilada. Se hoje em dia existem alguns artefactos para tornar o *Javascript* (principal linguagem de *scripting* associada ao HTML) mais difícil de compreender e descodificar, continua no entanto a ser possível aceder ao código fonte por completo. Assim sendo, o Cocos2d-x sai vencedor desta escolha.

Outro motivo por que recai a opção sobre o Cocos2d-x e me levou a inclui-la obrigatoriamente na análise às várias disponíveis, é o fato de a Cnotinfor ter já algum *software* desenvolvido nesta Framework e os seus colaboradores terem também por isso, conhecimento sobre esta.

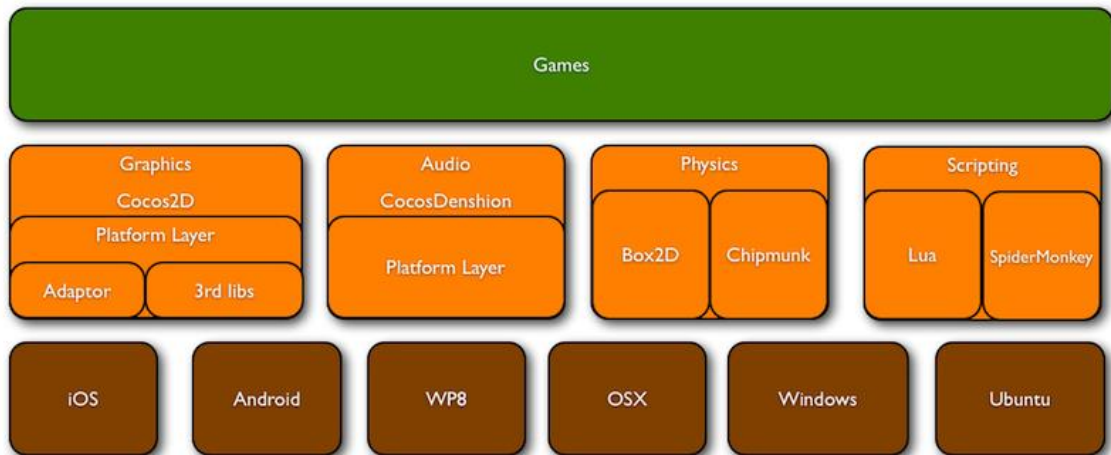


Figura 8 - Arquitetura da Framework Cocos2d-x [8]

5.1.4. TopQX

A Cnotinfor tem vindo a desenvolver e a converter recentemente, as suas aplicações para dispositivos não só fixos, como era comum, mas também móveis, permitindo assim que um utilizador possa adquirir uma licença de *software* e corrê-lo em múltiplas plataformas, para assim cumprir o lema desta Framework "Todos Podem Aprender a Qualquer Hora, em Qualquer Lugar".

Para tal, criou o projeto TopQX, que visa a criação de ferramentas de desenvolvimento de *software* multiplataforma de uma forma rápida e adaptável, através da criação de bibliotecas transversais aos vários projetos, assentando sobre a Framework Cocos2d-x.

Atualmente a TopQX contém já imensas classes e métodos comuns a quase qualquer aplicação, como formulários de login melhorados, botões com e sem texto, listas com *scroll*, métodos de posicionamento, etc., pelo que poupa imenso trabalho no desenvolvimento de novas aplicações.

É portanto uma Framework melhorada, desenvolvida a partir do Cocos2d-x, e não uma concorrente à mesma ou a outra ferramenta.

5.2. Base de dados local

A base de dados é implementada usando a biblioteca C++ SQLite, por ser extremamente leve em termos de recursos computacionais e não necessitar de servidor ou de configuração. É o método mais usado com a Cocos2d-x, no que se refere ao uso de bases de dados relacionais, pelo que existe uma extensa documentação *online* sobre o uso desta biblioteca.

5.3. Base de dados remota

Para o servidor remoto, onde os dados de todos os utilizadores do ActiveBrain são guardados, foi escolhida a implantação de uma base de dados MySQL num servidor da Cnotinfor, pelo fato de a base de dados seguir um modelo relacional, pela moderada quantidade de dados a guardar e, por satisfazer as nossas necessidades em termos funcionalidades e velocidade.

Sendo o ActiveBrain uma aplicação para um público muito específico, não é esperada uma concorrência de acessos e um crescimento de dados que justifique grandes preocupações de escalabilidade.

5.4. API REST

Esta API foi desenvolvida usando a linguagem de programação *server-side*, PHP e, o módulo `mod_rewrite` do servidor Apache, que permite transformar URLs apontando para um ou vários scripts PHP que consoante o URL, recebem diferentes parâmetros.

A PHP é de distribuição gratuita, fácil implementação e com um vasta comunidade e documentação *online*. É uma linguagem bastante madura e com uma enorme panóplia de funcionalidades e que interage inclusive com os bancos de dados MySQL, de forma bastante prática e perceptível.

Capítulo 6 Desenvolvimento

Enquanto o primeiro semestre foi quase integralmente dedicado ao estudo das tecnologias, primeiros testes e *Hello Worlds*, o segundo semestre envolveu já o desenvolvimento e implementação das diversas tecnologias que envolvem o ActiveBrain, nomeadamente aplicação multiplataforma, bases de dados e REST API.

O estagiário não se limitou ao estudo de tecnologias e bibliotecas existentes, como efetuou modificações à biblioteca cocos2d-x e à biblioteca TopQX para se adequarem às necessidades das interfaces projetadas. Estas alterações e inovações foram inclusive úteis para os restantes projetos da empresa e recebidas com entusiasmo pela equipa de desenvolvimento da Cnotinfor.

Devido a alterações significativas no conceito do ActiveBrain, que implicou modificação de interfaces, de estrutura de dados e da API, o projeto sofreu um significativo atraso em relação ao inicialmente planeado, pois a introdução de diversas novas interfaces e funcionalidades consideradas prioritárias em relação a outras planeadas, levou a que a implementação do jogo a incluir nesta fase, se iniciasse apenas em Junho.

Todo este atraso implicou não alcançar todas as metas inicialmente projetadas para este estágio.

Neste capítulo será descrito todo o trabalho desenvolvido e sua cronologia, resumindo o 1º semestre pois foi já alvo de relatório anterior, e comparando com a calendarização inicial definida.

6.1. Calendarização das tarefas

6.1.1. 1º Semestre

Como foi já referido, o 1º semestre foi quase totalmente dedicado ao estudo das tecnologias envolventes, da arquitetura a seguir e do Estado da Arte. Houve ainda lugar, após o estudo e leituras necessárias, ao desenho da arquitetura geral do ActiveBrain e da aplicação.

Segue-se o diagrama de Gantt relativo a este período, resumindo assim a calendarização cumprida:

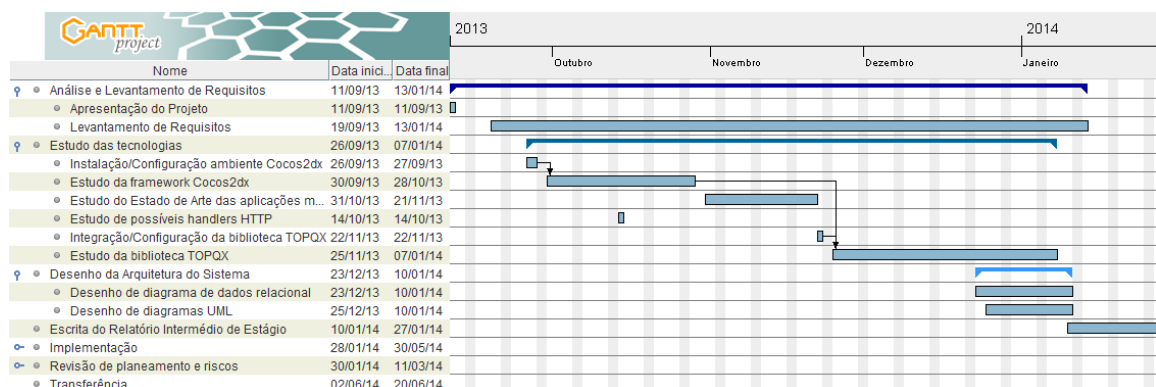


Figura 9 - Diagrama de Gantt relativo ao 1º semestre de Estágio

6.1.2. 2º Semestre

Com o 2º semestre veio a implementação de uma API REST, a criação de bancos de dados remoto e locais e a aplicação multi-plataforma.

Já com conhecimento sobre a biblioteca Cocos2d-x e algumas aplicações criadas para teste de funcionalidades, o estagiário partiu para o desenvolvimento da aplicação convicto de que a parte de desenvolvimento do jogo, mesmo sem a totalidade das funcionalidades especificadas, seria das mais dispendiosas do desenvolvimento do ActiveBrain. Tal não se sucedeu, tendo sido a plataforma quem ocupou mais tempo, não só pelas alterações de especificação que se vieram a dar a meio do semestre, mas também pela falta de capacidade do Cocos2d-x em implementar alguns componentes necessários, como por exemplo um scroll de conteúdos de fácil implementação e interação, limitações nas caixas de introdução de texto, etc. Isto levou a que algumas interfaces onde foram detetadas essas necessidades em falta na framework, demorasse mais do que o previsto.

Abaixo apresentam o Diagrama de Gantt relativo à calendarização prevista para o 2º semestre, e logo de seguida o Diagrama de Gantt da calendarização efetiva. Os diagramas encontram-se propositadamente aqui apresentados em tamanho reduzido para uma melhor comparação (em anexo encontram-se os Diagramas de Gantt em tamanho superior para possível análise mais detalhada):

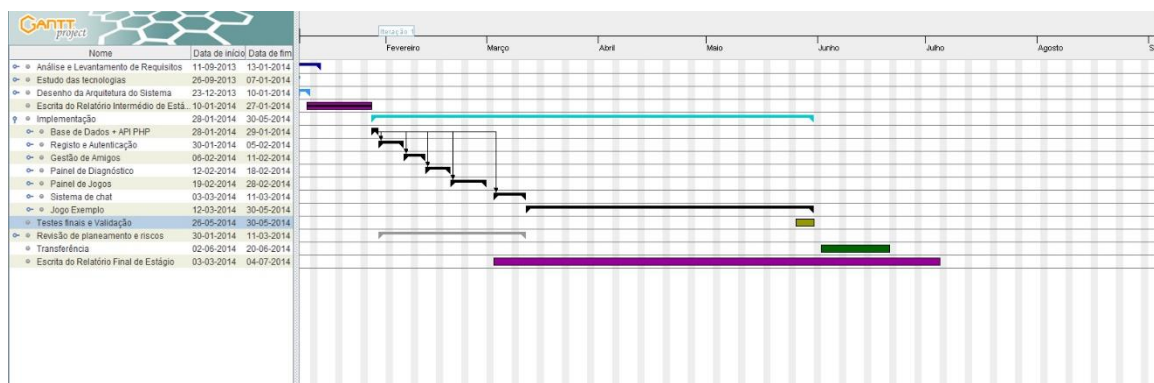


Figura 10 - Diagrama de Gantt previsto para o 2º semestre de Estágio, no período de 27 de Janeiro a 4 de Julho

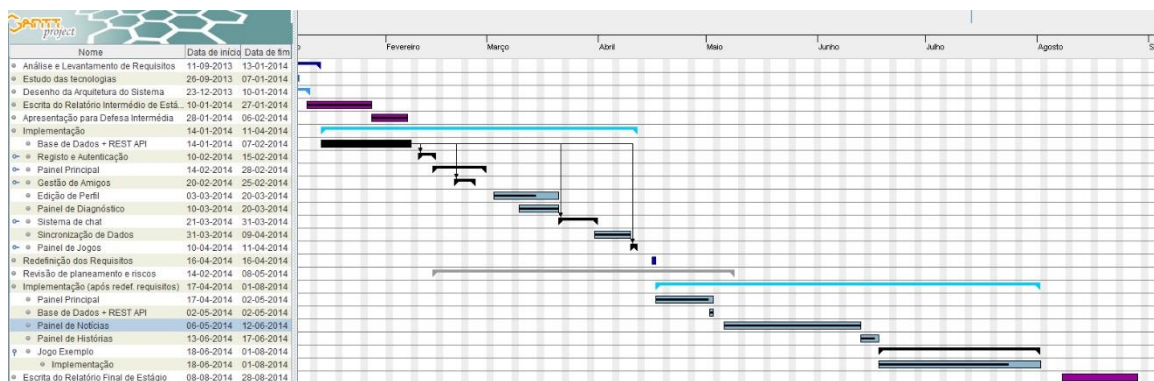


Figura 11 - Diagrama de Gantt efetivo do 2º semestre de Estágio, no período de 27 de Janeiro a 28 de Agosto

Através destes 2 diagramas percebe-se que a calendarização do 2º semestre sofreu bastantes desvios. Devido às alterações de requisitos que obrigaram à alteração de interfaces já criadas, à implementação de outras inexistentes e até à redefinição da estrutura de dados e da API REST, o estagiário divide aqui a implementação em 2 fases distintas, como se de duas versões da aplicação se tratassem.

6.2. Metodologia

O modelo de desenvolvimento de Software do projeto do ActiveBrain, foi escolha do estagiário, e dada a pequena dimensão da equipa e do próprio projeto, escolheu um modelo baseado em RUP, por considerar que este modelo seria o mais aconselhado. No entanto o modelo RUP não foi a ser seguido à risca (i.e., houve muita documentação UML descartada) pelo motivo referido.

Este projeto desde o seu início, passou pelas seguintes fases:

Análise e Levantamento de Requisitos

Nesta fase foi percebido o objetivo da aplicação, tanto no que toca ao seu uso e Público-Alvo, como aos fatores diferenciadores em relação a outras existentes. Foi especificada cada componente passo a passo e definidos os seus requisitos funcionais (interface, especificação do layout, etc.) e não funcionais (persistência de dados, requisitos de segurança, etc.). O ActiveBrain teve no entanto uma nova fase de levantamento de requisitos a meio do 2º semestre, quando a equipa envolvida no ActiveBrain decidiu efetuar algumas alterações ao conceito da plataforma, introduzindo novas funcionalidades, incluindo partilha de histórias e leitura de notícias.

Desenho da Arquitetura do Sistema e Escolha das Tecnologias

Levantados os requisitos funcionais e não-funcionais, foi desenhada a arquitetura do sistema de modo a responder aos requisitos funcionais e não-funcionais. Escolheram-se depois as tecnologias de implementação, as quais foram estudadas e testadas para conhecimento do estagiário e desfazer dúvidas levantadas quanto à capacidade das mesmas em responder a determinadas necessidades.

Implementação

Esta foi a fase de codificação e de criação de todas as componentes envolvidas no sistema, desde bases de dados, API, classes, etc. É uma fase que pode ser chamada diversas vezes, pois os testes podem levantar novos requisitos ou detetar falhas que impliquem reimplementação ou alterações ao código, à estrutura de dados, etc.

Testes finais e Validação

Cada interface implementada foi sujeita a testes funcionais pelo próprio estagiário. Algumas interfaces foram validadas pela responsável do projeto, Margarida Silva e apresentadas em reunião. Não foram no entanto feitos testes a todas as interfaces, nem feitos testes de usabilidade.

6.3. Gestão de riscos

No primeiro semestre foram calculados os riscos a que este projeto poderia estar exposto, e avaliados o seu impacto, probabilidade e consequente exposição.

Efetivamente, o risco com mais exposição (Derrapagem excessiva no tempo de implementação da fase pré-jogo) sucedeu-se, e o seu plano de contingência foi ativado. Foi por isso deixado para posterior implementação, todas os requisitos do tipo Pouco Importante ou Nada Importante, como por exemplo as funcionalidades inerentes à troca de mensagens entre

psicoterapeutas/médicos/acompanhante e o idoso e edição de fotografia. Também o plano de contingência relativo ao risco “Dificuldades na criação de mapas com grau de dificuldade dinâmico”, devido ao estagiário ter percebido que este risco seria inevitável, e assim criados mapas fixos.

Vejam os riscos calculados no início do projeto, ordenados por grau de exposição, do mais exposto para o menos exposto:

Risco	Impacto	Probabilidade	Exposição
Derrapagem excessiva no tempo de implementação da fase pré-jogo	Atraso das metas adiante. Atraso na conclusão global do projeto ou não conclusão do mesmo	80%	Alta
Plano de redução de risco	Ao final de cada dia estimar o tempo de conclusão da tarefa para antecipar uma possível derrapagem temporal		
Plano de contingência	Reavaliar a complexidade do jogo de modo a reduzir o tempo de implementação. Congelar requisitos Nada Importantes, Pouco Importantes ou mesmo Neutros		

Risco	Impacto	Probabilidade	Exposição
Complexidade de programação do jogo	Atraso na conclusão global do projeto ou não conclusão do mesmo	50%	Significativa
Plano de redução de risco	Usar sempre que possível bibliotecas/métodos pré-concebidos, que implementem as funcionalidades requeridas. Reutilizar código		
Plano de contingência	Reavaliar a complexidade do jogo de modo a reduzir o tempo de implementação. Congelar requisitos Nada Importantes e ou Pouco Importantes		

Risco	Impacto	Probabilidade	Exposição
Dificuldades de implementação por falha no conhecimento da linguagem	Atraso na conclusão global do projeto ou não conclusão do mesmo	40%	Significativa
Plano de redução de risco	Usar sempre que possível e adequado ao projeto, abordagens menos complexas		
Plano de contingência	Reforçar o conhecimento com leituras e tutoriais, com horas extra para não derrapar no tempo estimado de implementação. Pedir ajuda a elementos da empresa externos ao projeto. Congelar requisitos Nada Importantes e Pouco Importantes		

Risco	Impacto	Probabilidade	Exposição
Perda de dados	Retrocesso na timeline do projeto. Atraso de todas as metas adiante. Atraso na conclusão global do projeto ou não conclusão do mesmo	20%	Moderada
Plano de redução de risco	Fazer commits e pushes regulares (sempre que as alterações forem significativas) ao repositório local e remoto para ter redundância de dados		
Plano de contingência	Congelar requisitos Nada Importantes, Pouco Importantes ou mesmo Neutros		

Risco	Impacto	Probabilidade	Exposição
Dificuldades na criação de mapas com grau de dificuldade dinâmico	Redução da adaptabilidade do jogo	40%	Secundária
Plano de redução de risco	Estudo prévio da estrutura de dados a aplicar aos mapas e da representação gráfica. Envolver outros membros da empresa no brainstorming para procurar conhecimento que resolva o problema		
Plano de contingência	Reunir com Margarida Silva e definir x níveis de dificuldade com mapas estáticos		

Risco	Impacto	Probabilidade	Exposição
Elevadas alterações de requisitos durante os testes	Atraso na conclusão global do projeto, conclusão com funcionalidades diferentes dos requisitos finais ou não conclusão do mesmo	20%	Secundária
Plano de redução de risco	Rever os requisitos e definir bem quais dos requisitos são prioritários e essenciais		
Plano de contingência	Reunir com Nuno na tentativa de envolver mais elementos da empresa na alteração aos requisitos solicitados. Aplicar esforço extra no projeto com horas extraordinárias.		

Risco	Impacto	Probabilidade	Exposição
Problemas na exportação da aplicação para os diversos dispositivos	Falha na conclusão do projeto e sua aplicação no mercado	10%	Secundária
Plano de redução de risco	Basear código na biblioteca Cocos2d-x de modo a não fugir da framework interoperável		
Plano de contingência	Congelar requisitos Nada Importantes, Pouco Importantes ou mesmo Neutros. Reunir com Nuno na tentativa de envolver outros elementos da empresa na resolução do problema, nomeadamente os que têm alguma experiência neste processo		

Apesar de um dos fatores de atraso do projeto ter sido previamente calculado, foram também outros fatores não previstos que em parte, implicaram o atraso do projeto. As alterações de requisitos que o projeto sofreu, implicaram uma alteração significativa na implementação da aplicação. Porém, a importância destes requisitos adicionais na valorização do ActiveBrain, levou a que tais funcionalidades não fossem postas em espera, e fossem logo implementadas, levando contudo, a que outros requisitos iniciais fossem para um plano secundário.

6.4. Controlo de versões e backup de dados

A criação de versões do desenvolvimento parcial do projeto, é uma arma importante para poder anular erros, implementações mal feitas ou mesmo perda ou corrompimento de dados. Assim, durante toda a fase de implementação foi usado o sistema de controlo de repositório Git.

O Git permite o controlo de versões em repositório local e/ou remoto. Localmente sempre que houve alterações significativas, foi atualizado o repositório Git, evitando possíveis erros não recuperáveis pelo IDE.

No servidor da Cnotinfor, foram criados 2 repositórios: Um para a aplicação, outro para a API REST.

O repositório remoto foi carregado com uma nova versão, sempre que algum módulo era finalizado, ou se o estagiário decidia que tinham sido feitas demasiadas alterações sem haver sincronização entre repositórios local e remoto.

6.5. Documentação

Uma das primeiras tarefas do estagiário neste projeto, foi iniciar a criação de uma série de documentação e diagramas UML, de modo a conseguir entender o seus dados, métodos de comunicação, requisitos, etc., e mais tarde o ActiveBrain possa ser entendido e alterado por outros colaboradores da empresa.

Assim, foi criada pelo estagiário, a seguinte documentação:

- Descrição da Aplicação
- Análise de Requisitos
- Especificação e requisitos da Base de Dados
- Diagramas Entidade-Relacionamento de dados do dispositivo e servidor
- Diagrama de Casos de Uso da Aplicação
- Diagramas de Sequência para as diversas funcionalidades
- Manual de utilização da API REST
- Protótipos de mapas de jogo

6.6. Progresso de desenvolvimento e desvios ao planeamento

Como foi referido atrás, o ActiveBrain sofreu alterações de especificação a meio do 2º semestre, obrigado à redefinição de estruturas de dados, de interfaces e à reimplementação de algum código já terminado.

Estas alterações fizeram com que outras funcionalidades fossem adiadas no planeamento, por serem tanto ou menos importantes que as novas funcionalidades definidas.

Assim, chegado o fim do estágio, o ActiveBrain tem a maioria das suas funcionalidades desenvolvidas, não estando no entanto validadas na sua totalidade, pelo que poderão ainda surgir algumas alterações a efetuar fruto de falhas encontradas ou defeitos nos requisitos levantados. Outros requisitos, estão ainda por implementar, nomeadamente:

- Componente de socialização durante o jogo

O sistema de chat está implementado e testado, estando inclusivé em condições de dividir a tela com o jogo. Falta no entanto resolver alguns problemas entretanto encontrados, nomeadamente a sobreposição do teclado à area de jogo enquanto se escreve, ou o fecho da janela de chat com mensagens não enviadas quando o jogo é encerrado.

- *Feedback* de jogo

A estrutura de dados está implementada, bem como a API REST está já preparada para receber e devolver *feedback* relativo a determinado jogo e/ou determinado jogador. Não está no entanto ainda, desenvolvida qualquer interface de visualização e escrita.

- Notificação de mensagens recebidas

O ActiveBrain tem já um sistema de notificações implementado, que se sobrepõe a qualquer interface mostrando a mensagem/notificação recebida, seja mensagem ou convite para jogar. Falta no entanto o painel listando todas as mensagens.

- Sugestão de jogo para jogo individual

Estando ainda para já numa fase em que apenas um jogo está em desenvolvimento, esta funcionalidade não é prioritária pois não há escolha possível. Não está ainda implementada.

- Sugestão de jogo para jogo multijogador

Estando ainda para já numa fase em que apenas um jogo está em desenvolvimento, esta funcionalidade não é prioritária pois não há escolha possível. Não está ainda implementada.

Por opções da empresa, existem outros projetos estrategicamente mais prioritários que o ActiveBrain a serem desenvolvidos envolvendo todos a equipa da Cnotinfor. Por este motivo não se chegou ainda a uma fase de testes completa (às funcionalidades já implementadas), incluindo testes de usabilidade e qualidade, tendo o estagiário apenas efetuado testes funcionais ao completar cada módulo da plataforma.

O ActiveBrain continua a ser uma aplicação valorizada e que terá a sua conclusão em breve, sendo que o estagiário desenvolveu toda a documentação necessária para a sua fácil continuação, mesmo que venha a ser tarefa de outro colaborador.

6.7. Outras tarefas envolvendo o estagiário

Ao longo do ano o estagiário foi extraordinariamente bem acolhido na empresa, integrando as reuniões, opinando sobre o ActiveBrain e até sobre outros projetos e participando nas atividades de pessoal.

Existiram no entanto algumas atividades extra-projeto a destacar, que enriqueceram o conhecimento do estagiário e aumentaram o seu valor na empresa.

6.7.1. Implementação de duas novas classes na biblioteca TopQX

Despoletada pela necessidade para o ActiveBrain, o estagiário iniciou o desenvolvimento de uma nova classe que implementasse uma *scroll* semelhante a um elemento *scrollable* HTML, que permitisse assim remover e adicionar objetos gráficos automaticamente posicionados lado a lado ou na vertical, consoante parâmetros a definir.

Igualmente não existia ainda métodos para recorte de imagem, como por ex. fotografias mascaradas em círculo, pelo que o estagiário se debruçou sobre esta necessidade e implementou uma biblioteca que não só mascara imagens, como cria retângulos com cantos arredondados, com sombreado, com limites configuráveis, etc.

Ao perceber as vantagens destas inovações, outro membro da equipa de desenvolvimento da Cnotinfor, Joel Oliveira, reuniu-se com o estagiário para juntos criarmos uma biblioteca a integrar na TopQX para ser assim usada para futuros projetos.

6.7.2. Microsoft Developer Tech Refresh

A Cnotinfor foi convidada a participar no Microsoft Developer Tech Refresh, um evento anual que decorre na sede da Microsoft em Lisboa, que divulga e analisa as tendências de desenvolvimento atuais, principalmente vocacionando claro, produtos Microsoft.

O estagiário que já tinha por si mesmo algum interesse em participar, foi convidado e apoiado pela empresa a assistir a este evento, adquirindo assim conhecimento útil principalmente no que toca ao desenvolvimento de aplicações móveis.

6.7.3. Escrita de artigo para revista da empresa

A Cnotinfor tem uma revista *online*, a “Bica”, onde divulga as atividades da empresa, os projetos em que se encontra envolvida, as tendências e novidades na área da educação e da inclusão, etc.

O estagiário foi convidado a relatar a sua experiência enquanto colaborador da empresa, escrevendo assim um artigo que foi bastante elogiado pela direção da empresa e colegas de trabalho.^[27]

Capítulo 7

Conclusões e Trabalho Futuro

O ActiveBrain é um projeto com valor, não só por ser vocacionado para um público necessitado de intervenção e inovação na área do declínio cognitivo, mas também porque aproxima a população mais velha das novas tecnologias, nomeadamente da socialização *online* e do lazer tecnológico. Apesar de existirem aplicações no mercado que implementam parte das funcionalidades projetadas para esta plataforma, o ActiveBrain colmatará o mercado, principalmente o da língua portuguesa, na medida em que inclui incentivos à socialização, seja por interação direta entre os utilizadores, pela partilha de histórias de vida ou simplesmente pela competição saudável e pela partilha de experiência de jogo.

O projeto irá em breve ser terminado nesta versão inicial, para ser depois revisto, testado e iniciada a sua integração no mercado. Será no entanto necessário o desenvolvimento de outros jogos, que se foquem na estimulação de outros eixos de progresso cognitivo, como a atenção.

Importa também em trabalho futuro, pesquisar novas formas de interação social que não sejam limitados à escrita em teclado, pois para a maioria do público idoso a introdução de palavras usando um teclado é uma tarefa morosa e nem sempre motivadora.

Quando se trata de desenvolver aplicações para públicos específicos, o conhecimento das suas necessidades, expectativas e limitações, é de extrema importância e pode significar requisitos específicos na arquitetura do sistema, pelo que nenhum detalhe deve ser deixado ao acaso aquando do seu estudo. A população de terceira idade é um desses casos, e se muitos tendem a preocupar-se apenas com o tamanho dos textos e dos botões, a verdade é que as limitações fisiológicas do idoso vão muito para além da dimensão dos elementos de uma interface.

O estágio foi bastante rico no que toca à aprendizagem do estagiário e à valorização pessoal no mundo do trabalho, sendo este reconhecido pela empresa e, até, alvo de proposta de trabalho futuro.

No entanto houve falhas cometidas, que deverão ser agora analisadas para que no futuro não impliquem atrasos no *time-to-market*, principalmente quando se trate de aplicações que integram a base financeira da empresa. Estas falhas incluem o levantamento incompleto de requisitos, e, acima de tudo, o tempo perdido no desenvolvimento de classes novas que poderiam ser substituídas por alternativas já desenvolvidas e, mais tarde com uma primeira versão já lançada, aí sim melhoradas ou reinventadas.

Foi no entanto acima de tudo, na ótica do estagiário, uma experiência bastante positiva, em que ficou a conhecer melhor o desenvolvimento de aplicações móveis, principalmente com foco na interoperabilidade de dispositivos e sistemas operativos, o desenvolvimento de aplicações para inclusão social e claro, o mundo do trabalho.

Referências

- [1] SANTANA, ISABEL. 2013. Especial Neurologia. Jornal Médico N.º6, Edições Newsfarma
- [2] LINDEBOOM R, SCHMAND B, HOLMAN R, DE HAAN RJ, VERMEULEN M., 2004. Improved brief assessment of Cognition in aging and Dementia, Department of Clinical Epidemiology, Amsterdam
- [3] Lumosity - <http://www.lumosity.com> [Último acesso a 21-Jan-2014]
- [4] Fit Brains Trainer - <http://www.fitbrains.com> [Último acesso a 22-Jan-2014]
- [5] Cérebro Melhor - <http://www.cerebromelhor.com.br> [Último acesso a 21-Jan-2014]
- [6] Cognifit - <http://www.cognifit.com> [Último acesso a 21-Jan-2014]
- [7] Unity - <http://unity3d.com/> [Último acesso a 24-Jan-2014]
- [8] Cocos2d-x - <http://www.cocos2d-x.org/> [Último acesso a 24-Jan-2014]
- [9] Marmalade - <https://www.madewithmarmalade.com/> [Último acesso a 24-Jan-2014]
- [10] RUP- <http://www.wthreex.com/rup/portugues/index.htm> [Último acesso a 22-Jan-2014]
- [11] SILVA, MARGARIDA LOPES. 2014. Relatório Final de Estágio - Plataforma de Ginástica Cerebral - ActiveBrain4All. Universidade de Trás-os-Montes e Alto Douro
- [12] Vita Health (agregador de conteúdos) <http://www.vitalhealth.pt/senior> [Último acesso a 25-Jan-2014]
- [13] Alzheimer Portugal - <http://www.alzheimerportugal.org/> [Último acesso a 22-Jan-2014]
- [14] Wiki VS - http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL [Último acesso a 26-Jan-2014]
- [15] Exame Informática - http://exameinformatica.sapo.pt/melhores-apps/apps_android/2014-01-22-Melhores-apps-Treine-o-seu-cerebro [Último acesso a 22-Jan-2014]
- [16] Alzheimer's Disease International, 2009 - Relatório sobre a Doença de Alzheimer no Mundo.
- [17] MENDONÇA A. - Envelhecimento Ativo e Educação ao Longo da Vida - Centro de Competência de Ciências Sociais - <http://www3.uma.pt/alicemendonca/1FCSH.pdf> [Último acesso a 08-Abr-2014]
- [18] PHIRIYAPOKANON, TANID – “Is a big button interface enough for elderly users?”, 2011. Mälardalen University, Sweden
- [19] MEHRBA M., ASIF M. 2010. Challenges and Strategies in Mobile Phones Interface for elder people – Master Thesis in Computer Science – Blekinge Institute of Technology, Sweden
- [20] Nolan D. Normal Age-Related Vision Loss and Related Services for the Elderly - Stephen F. Austin State University –

http://www.laurenscharff.com/research/donia/aging_visual_changes.htm [Último acesso a 12-Ago-2014]

[21] TECMUNDO - <http://www.tecmundo.com.br/sistema-operacional/60596-ios-android-windows-phone-numeros-gigantes-comparados-infografico.htm> [Último acesso a 19-Ago-2014]

[22] Hypertext Transfer Protocol HTTP/1.1, World Wide Web Consortium (W3C) - <http://www.w3.org/Protocols/rfc2616/rfc2616.html> [Último acesso a 21-Ago-2014]

[23] Associação Rede de Universidades da Terceira Idade - <http://www.rutis.org/> [Último acesso a 21-Ago-2014]

[24] APOSENIOR - <http://www.aposenior.pt/> [Último acesso a 21-Ago-2014]

[25] Associação Portuguesa de Psicogerontologia – <http://www.app.com.pt> [Último acesso a 21-Ago-2014]

[26] IEEE Recommended Practice for Software Requirements Specifications - IEEE Computer Society - <http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf> [Último acesso a 22-Ago-2014]

[27] BICA 176 – Imagina/Cnotinfor newsletter - <http://bica.imagina.pt/2014/a-cnotinfor-nos-meus-primeiros-passos-e-nos-ultimos>

ANEXO A – Requisitos Funcionais da aplicação

Lista de Requisitos Funcionais da Aplicação, avaliados por um grau de importância (Nada importante, Pouco importante, Neutro, Importante, Muito Importante) e Diagramas de Casos de Uso da área (interface) correspondente.

Painel de Autenticação

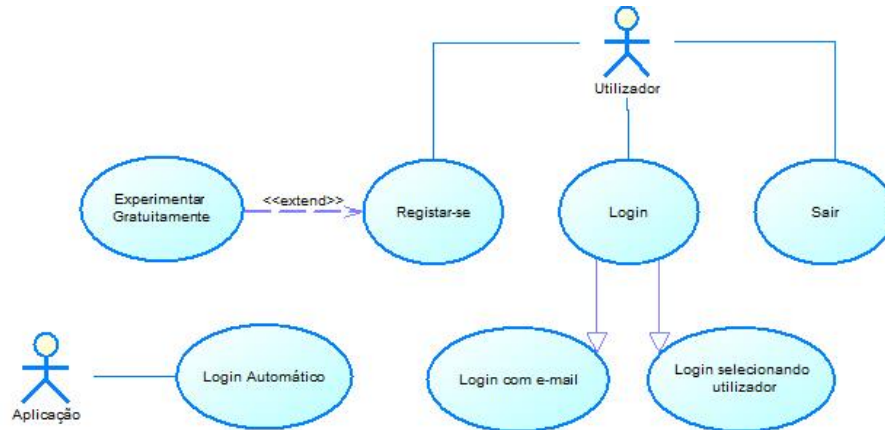


Figura 12- Diagrama de Casos de Uso para o “Painel de Autenticação”

[RF101] Login com e-mail

A aplicação deverá ter um sistema de autenticação por inserção de e-mail e palavra-chave. A inserção dos dados será feita por caixas de introdução de texto e botão para submeter dados. A password deverá conter a opção de ficar visível, ou mascarada com asteriscos (*), sendo que o último caratér deverá ser visível durante a sua inserção, desaparecendo após 1 segundo de ser inserido ou quando se escreva um caratér a seguir. Em caso de erro nos dados, o utilizador deverá ser informado por meio de uma mensagem de erro.

Pré-condições: Aplicação ter iniciado. Não haver sessão ativa.

Prioridade: Muito Importante

[RF102] Login automático

Quando o último utilizador tiver encerrado a aplicação sem ter feito *logout* do sistema, a plataforma ao reiniciar deverá iniciar o seu perfil automaticamente sem perguntar por utilizador e/ou password e apresentar de imediato o Menu Principal. Esta ação deverá ser completamente transparente para o utilizador.

Pré-condições: Aplicação ter iniciado. Não haver sessão ativa. Último utilizador não efetuou *logout*

Prioridade: Importante

[RF103] Login selecionando utilizador

O dispositivo deverá recordar os últimos 3 utilizadores. A aplicação ao iniciar, caso o último utilizador tenha feito *logout* da aplicação, então deverão aparecer por ordem de último acesso,

os últimos (até ao máximo de 3) utilizadores com nome e fotografia (se disponível). Ao clicar sobre um, a plataforma apenas pedirá para inserir a password. A password deverá conter a opção de ficar visível, ou mascarada com asteriscos (*), sendo que o último caratér deverá ser visível durante a sua inserção, desaparecendo após 1 segundo de ser inserido ou quando se escreva um caratér a seguir. Em caso de erro na password, o utilizador deverá ser informado por meio de uma mensagem de erro.

Pré-condições: Aplicação ter iniciado. Não haver sessão ativa. Haver utilizadores guardados em memória local (já houve *logins* no dispositivo anteriormente)

Prioridade: Neutro

[RF111] Registrar-se

É necessário uma interface de registo, para que um novo utilizador possa criar uma conta pessoal inserido os seus dados pessoais (e-mail, palavra-chave e confirmação de password, nome, data de nascimento e número de série). Deverá ser informado por meio de uma mensagem de erro, caso o e-mail já exista ou caso a password e respetiva confirmação não coincidam. A password deverá conter um mínimo de 4 caratères alfa-numéricos.

Pré-condições: Aplicação ter iniciado. Não haver sessão ativa.

Prioridade: Importante

[RF112] Experimentar gratuitamente

Durante o registo, o utilizador poderá optar por, ao invés de inserir um número de série válido, selecionar gratuitamente a opção de experimentar a aplicação durante 30 dias, tendo que selecionar para isso uma *checkbox* no formulário de registo.

Pré-condições: Aplicação ter iniciado. Não haver sessão ativa. Registo efetuado sem nº de série.

Prioridade: Pouco importante

[RF121] Sair

O utilizador deverá ter a opção de encerrar a aplicação. Para isso deverá existir um botão para o efeito, que o utilizador ao carregar sobre ele, a aplicação deverá pedir uma confirmação informando este que clicou na opção de sair. O utilizar escolherá então “Sim” para confirmar o encerramento da aplicação, que deve acontecer logo de seguida, ou “Não” para fechar a mensagem e continuar o uso normal da aplicação.

Pré-condições: Aplicação ter iniciado. Não haver sessão ativa. Registo efetuado sem nº de série.

Prioridade: Pouco importante

Menu Principal

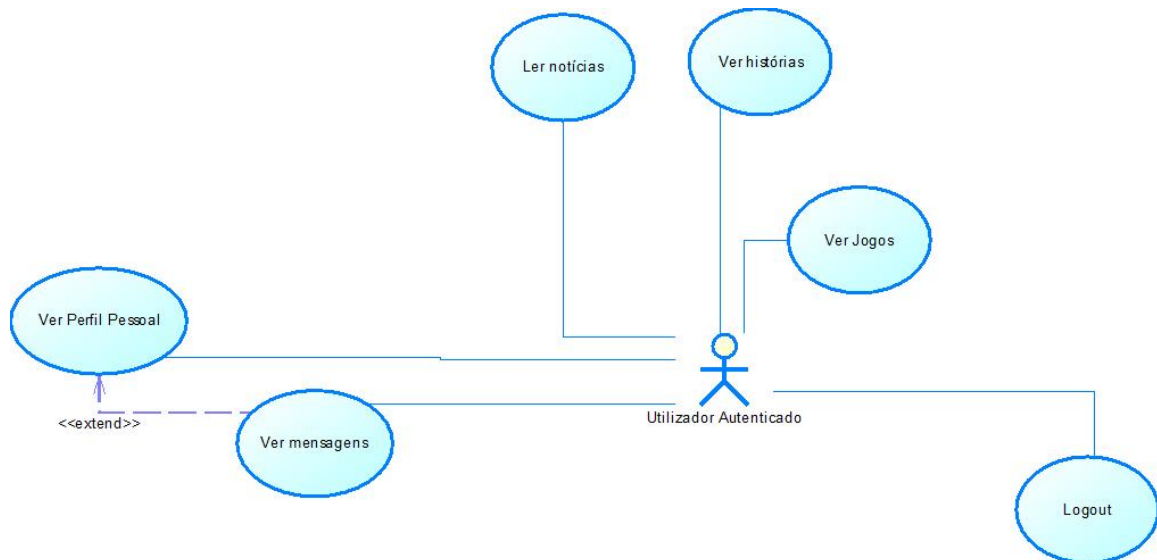


Figura 13 - Diagrama de Casos de Uso para o "Menu Principal"

[RF201] Escolha de funcionalidade (atualizado no 2º semestre)

O Menu Principal deverá conter 3 botões em destaque, que dêem acesso à área de jogos, de notícias e de histórias de utilizador. Deverá ainda, dar acesso à Edição de Perfil e à opção de fazer *Logout*.

Pré-condições: Aplicação ter iniciado e *login* efetuado.

Prioridade: Muito importante

[RF203] Logout

O utilizador deverá ter disponível um botão de *logout* onde ao clicar nele, a aplicação deverá pedir confirmação através de mensagem com opções "Sim" e "Cancelar". Ao clicar em "Sim" a sessão deverá encerrar e aplicação redirecionar o utilizador para o menu de Painel de Autenticação. Ao clicar "Cancelar" a janela de confirmação deve encerrar e nada mais acontecer.

Pré-condições: Aplicação ter iniciado e *login* efetuado.

Prioridade: Neutro

[RF202] Notificação de mensagens

Ao entrar no Menu Principal, o utilizador deverá ser notificado, através de uma mensagem associada a um tutor, ou simplesmente um painel de mensagens que surja, de que tem uma nova mensagem por ler. Ao clicar sobre o painel, deverá abrir de imediato o painel de mensagens com a nova mensagem visível.

Pré-condições: Aplicação ter iniciado e *login* efetuado. Haver mensagens por ler.

Prioridade: Nada importante

Perfil de Utilizador

O acesso ao Painel de Utilizador é resultado do requisito RF201 e como tal implica que a aplicação tenha previamente iniciado e o utilizador iniciado sessão (*login*). Como tal, as pré-condições deste módulo assumem a validação das condições de acesso a ele.

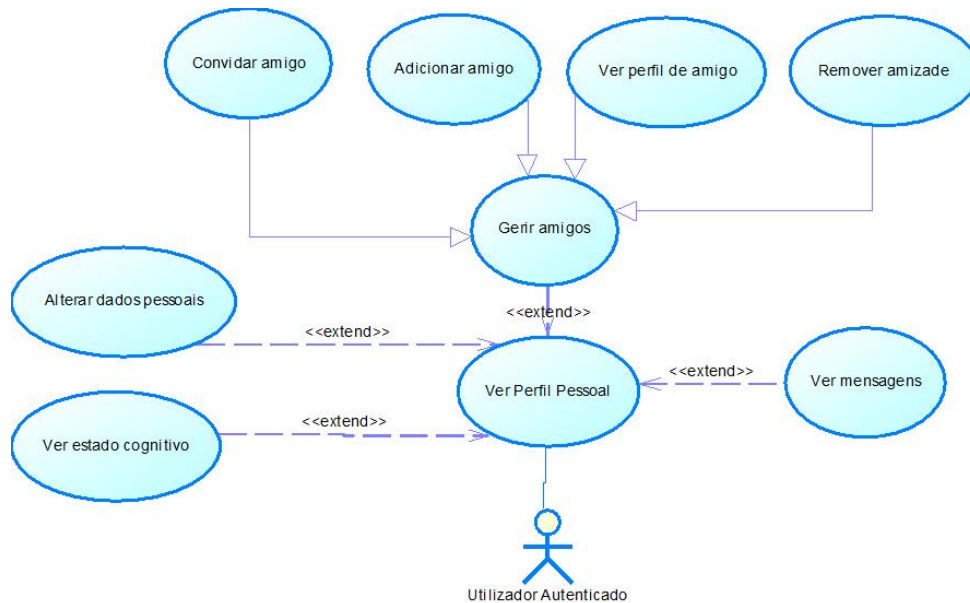


Figura 14 - Diagrama de Casos de Uso para "Perfil de Utilizador"

[RF301] Edição de perfil

O utilizador terá acesso aos seus dados de perfil, podendo alterar o seu e-mail e a sua password. Deverá ser informado por mensagem de erro, caso o e-mail já exista ou caso a password e respetiva confirmação não coincidam. Poderá alterar também o nome de apresentação.

Prioridade: Importante

[RF302] Edição de foto

Os utilizadores poderão ter uma fotografia associada. O sistema deverá permitir adicionar uma foto ao perfil pessoal e alterá-la sempre que o utilizador assim queira, tendo para tal 2 botões, um para seleção de ficheiro local, outro para captura direta através da câmara do dispositivo. O sistema deverá alterar a fotografia assim que for submetida, sem necessidade de passos adicionais.

Prioridade: Pouco Importante

[RF311] Ver amigos

Deverá existir uma listagem dos atuais amigos, mostrando fotografia e nome. Ao clicar sobre cada um, deverá ser mostrada a informação básica de perfil, nomeadamente nome, fotografia e pontuação geral, bem como a opção de o remover como amigo por meio de um botão de remoção, que dará lugar a uma mensagem de confirmação quando nele se clicar. Ao voltar a clicar, o amigo será removido. Caso o utilizador em questão esteja conectado, deverá ser visível uma opção para convidar o amigo para jogar a dois.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: Importante

[RF312] Pesquisar pessoas

Deverá existir um formulário de pesquisa de pessoas (por nome ou por e-mail). Ao clicar sobre o botão de submissão de pesquisa, deverá surgir uma listagem de resultados, ou uma

mensagem indicando nenhum resultado caso assim seja. Ao clicar sobre um dos resultados, deverá surgir uma mensagem de confirmação perguntando se o utilizador deseja adicionar essa pessoa como amiga, cuja confirmação (botões confirmar e cancelar) irá adicionar a pessoa respetiva aos seus amigos.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: **Importante**

[RF313] Convite a amigos

Quando na pesquisa de amigos for inserido um e-mail que não conste do registo de utilizadores, o utilizador atual deverá ter a possibilidade de enviar um convite por e-mail para que o amigo experimente a aplicação, bastando para isso clicar num botão.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: **Nada Importante**

[RF321] Ver mensagens

O profissional de saúde pode aconselhar o paciente através da plataforma, enviando mensagens pessoais que serão depois mostradas num painel de mensagens, como que uma caixa de entrada. O utilizador deverá ter acesso a uma lista ordenada de mensagens em que o seu título é mostrado, expandindo o seu conteúdo ao clicar sobre elas. Deverá ser mostrada a data da sua receção.

Pré-condições: Ter mensagens guardadas localmente ou ter mensagens guardadas remotamente e dispositivo conetado à Internet.

Prioridade: **Nada Importante**

[RF331] Ver diagnóstico

Este painel apresentará o progresso (os valores) dos eixos cognitivos Atenção, Memória, Flexibilidade, Velocidade e Resolução de Problema, e a pontuação geral. Os valores cognitivos, deverão ser mostrados por via de gráficos que demonstrem a sua evolução ao longo das últimas 10 sessões. A pontuação deverá ser mostrada sob a forma numérica.

Prioridade: **Muito Importante**

[RF341] Voltar ao Menu Principal

O utilizador deverá ter facilmente acessível, um botão para regressar ao Menu Principal bastando um só clique para o efeito, excepto se este estiver a editar os seus dados. Neste caso deverá ser perguntado ao utilizador se deseja guardar os dados, descartá-los ou cancelar a ação, mostrando de forma clara 3 botões com estas opções.

Prioridade: **Muito Importante**

Painel de Jogos

O acesso ao Painel de Jogos é resultado do requisito RF201 e como tal implica que a aplicação tenha previamente iniciado e o utilizador iniciado sessão (*login*). Como tal, as pré-condições deste módulo assumem a validação das condições de acesso a ele.

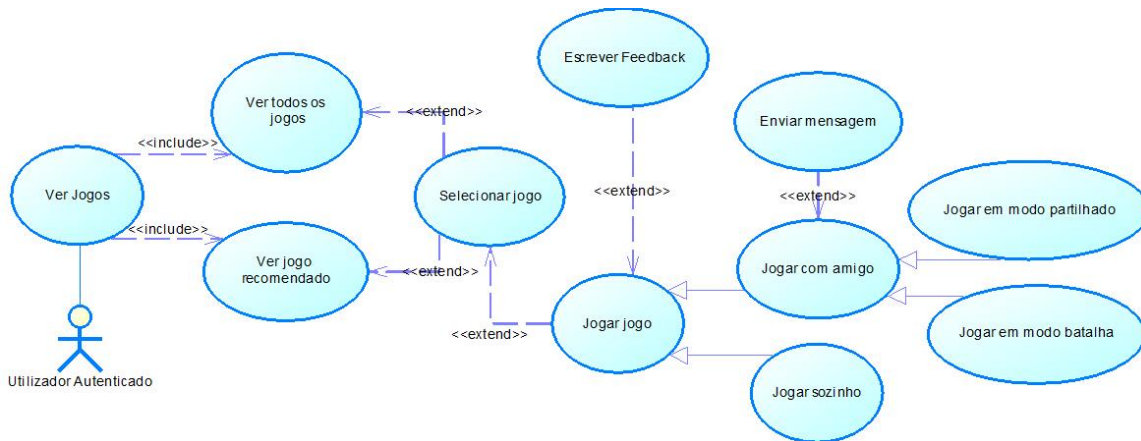


Figura 15 - Diagrama de Casos de Uso para o "Painel de Jogos"

[RF401] Listar jogos

Terá que existir um painel onde serão apresentados os jogos disponíveis, por categoria (áreas cognitivas) e também os adequados ao utilizador em questão. Será visível igualmente um resumo do diagnóstico do jogador, no que toca às áreas cognitivas a desenvolver, para que este saiba de jogos escolher.

Prioridade: Muito Importante

[RF402] Informação sobre o jogo

Ao clicar sobre o jogo a plataforma deverá mostrar uma pequena informação sobre o jogo, nomeadamente o nome do jogo, os objetivos do jogo, para que áreas cognitivas este está vocacionado e as mensagens de feedback de outros jogadores.

Prioridade: Importante

[RF403] Modo de jogo

Ao seleccionar um jogo, existirá ainda um botão para iniciar o jogo e outro para desafiar um amigo a jogar. Este botão ao carregar mostrará ainda 2 opções: Modo partilhado, ou modo batalha.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: Importante

Painel de Histórias

O acesso ao Painel de Histórias é resultado do requisito RF201 e como tal implica que a aplicação tenha previamente iniciado e o utilizador iniciado sessão (*login*). Como tal, as pré-condições deste módulo assumem a validação das condições de acesso a ele.

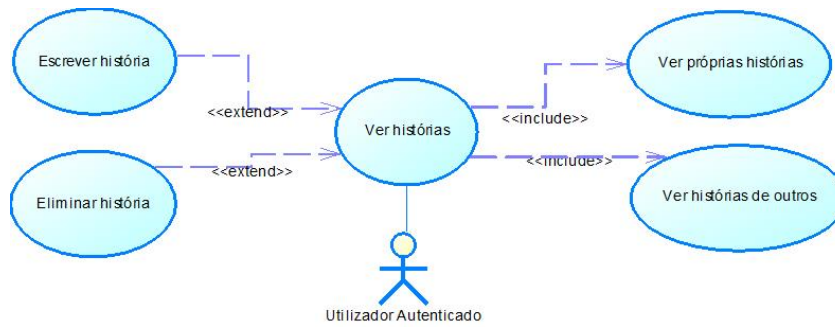


Figura 16 - Diagrama de Casos de Uso para "Ver Histórias"

[RF501] Listar histórias (atualizado no 2º semestre)

As histórias de toda a comunidade deverão aparecer numa interface própria, listadas por ordem de publicação, cada uma com um botão que explicita a sua função, abrir a história para leitura. Ao clicar sobre o respetivo botão, deverá ser expandida a história completa. As histórias escritas pelo próprio utilizador, deverão estar agrupadas numa zona diferente.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: Importante

[RF511] Escrever história (atualizado no 2º semestre)

O utilizador poderá escrever uma nova história para publicar, escrevendo um assunto e um corpo da história. Estes dados deverão ser possíveis de introduzir através de um formulário. A submissão da história será feita por meio de um botão com a sua função explícita. Ao clicar o sistema deverá validar o assunto e o corpo (invalidar campos em branco) e inserir a história caso os dados sejam válidos. Caso sejam inválidos, deverá ser mostrada mensagem de erro indicando o problema.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: Importante

[RF521] Eliminar história (atualizado no 2º semestre)

O utilizador poderá eliminar as suas histórias, tendo para isso que clicar num botão de eliminação, que estará visível junto ao título da história. Ao clicar, deverá ser pedida confirmação ao utilizador, através de uma mensagem e as opções "Confirmar" e "Cancelar". Ao clicar em "Confirmar" a história deverá ser eliminada e as suas histórias reordenadas caso fiquem espaços vazios entre elas consequência da eliminação. Caso clique em "Cancelar", a mensagem de erro deverá fechar e nada mais suceder.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: Importante

Painel de Notícias

O acesso ao Painel de Notícias é resultado do requisito RF201 e como tal implica que a aplicação tenha previamente iniciado e o utilizador iniciado sessão (*login*). Como tal, as pré-condições deste módulo assumem a validação das condições de acesso a ele.



Figura 17 - Diagrama de Casos de Uso para o "Ver notícias"

[RF601] Listar notícias (atualizado no 2º semestre)

As notícias retornadas pelo servidor, deverão ser listadas mostrando apenas o título, e ordenadas por data. Caso a listagem de títulos ultrapasse o tamanho da janela para eles destinados, estes deverão ser inseridos numa scroll. Cada título deverá ter um botão com a funcionalidade de leitura explícita. Ao clicar sobre qualquer um, a notícia deverá ser expandida por modo a mostrar todo o seu conteúdo incluindo possíveis imagens embutidas.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: Importante

Modo de jogo

O acesso ao Modo de Jogo implica o acesso prévio ao Painel de Jogos e o requisito RF403. Como tal, as pré-condições deste módulo assumem a validação prévia das condições de acesso a ele.

A aplicação deverá ter pelo menos um jogo completo, que inclui os seguintes requisitos funcionais:

[RF701] Instruções iniciais

O jogo deverá iniciar-se com um painel descrevendo quais os objetivos do jogo e como jogar. Este painel conterà apenas um botão para iniciar o jogo, não havendo qualquer outra interação.

Prioridade: Importante

[RF711] Janela de chat (para jogos multijogador)

Caso seja um jogo multijogador, a área de jogo deve ser dividida com uma janela de chat lateral, onde aparecerão por ordem cronológica e com identificação clara e de rápida perceção do remetente e destinatário, as mensagens trocadas.

Pré-condições: Jogo em modo multilijogador (ver RF403).

Prioridade: Muito Importante

[RF712] Caixa para envio de mensagem de chat (para jogos multijogador)

Caso seja um jogo multijogador, ao fundo da janela de chat deverá haver uma caixa de texto para escrever uma mensagem e um botão para a enviar.

Pré-condições: Jogo em modo multijogador (ver RF403).

Prioridade: Muito Importante

[RF721] Painel de resumo final em jogo singular

No final do jogo será apresentado um painel com a pontuação conseguida e com o progresso cognitivo conseguido nas 5 áreas. Será também apresentado um comparativo de pontuação com a média geral (a todos os jogadores) e com a pontuação máxima já conseguida pelo próprio jogador. Este painel deverá conter botões para voltar ao painel de jogos, ou jogar o nível seguinte.

Prioridade: Muito Importante

[RF731] Painel de resumo final singular em jogo multijogador

No final do jogo será apresentado um painel com a pontuação conseguida e com o progresso cognitivo conseguido nas 5 áreas. Será também apresentado um comparativo de pontuação entre os 2 jogadores, mostrando fotografia, nome e pontuação, enquanto que a janela de chat continuará visível e pronta a enviar novas mensagens. Este painel deverá conter botões para voltar ao painel de jogos, ou jogar o nível seguinte. Ao clicar no botão para voltar ao painel de jogos, a aplicação deverá pedir confirmação através de uma mensagem informando que esta ação irá desconetar os dois jogadores, e pedindo a confirmação através de um botão “Confirmar” ou cancelando através de outro botão (“Cancelar”).

Pré-condições: Jogo em modo multijogador (ver RF403).

Prioridade: Muito Importante

[RF741] Painel de resumo final - Comparativo com outros

Deverá ser visível também no painel de resumo final, a pontuação de outros jogadores, dando prioridade aos amigos da sua rede, mostrando fotografia, nome e pontuação, ordenada por pontuação do mais pontuado para o menos pontuado. Caso existam demasiados amigos que já jogaram o respetivo jogo, a aplicação deverá mostrar apenas os 5 mais pontuados.

Pré-condições: Dispositivo conetado à Internet ou haver registo de jogos de outros jogadores memorizados localmente.

Prioridade: Importante

[RF751] Sugerir jogo a amigo

No painel de resumo final, deverá ser possível sugerir a um amigo que jogue o respetivo jogo. Para isso, deverá aparecer a listagem dos amigos do utilizador, na qual este irá clicar sobre um amigo. Seguidamente o sistema envia uma mensagem ao amigo informando-o que o primeiro lhe sugeriu um jogo. O sucesso no envio da mensagem deverá ser confirmado ao remetente através de uma notificação de sucesso.

Pré-condições: Dispositivo conetado à Internet.

Prioridade: Nada Importante

[RF761] Feedback

No painel de resumo final, deverá ainda visível um painel de feedback onde serão mostradas as mensagens de opinião relativas a este jogo, ordenadas por data de publicação, dadas por todos os jogadores que já o jogaram. Estas mensagens serão mostradas anexadas à fotografia e nome do jogador. Deverá ser possível ainda o jogador introduzir a sua mensagem de opinião, através de uma pequena caixa de introdução de texto.

Pré-condições: Dispositivo conetado à Internet ou haver registo de jogos de outros jogadores memorizados localmente. Para escrever feedback não tem que existir conexão.

Prioridade: *Importante*

[RF771] Jogo seguinte

No final do jogo deverá ser visível também uma opção para continuar a jogar aumentando o grau de dificuldade. Esta opção será selecionada por meio de botão claramente identificado.

Prioridade: *Importante*

Jogo específico – “Qual o melhor caminho”

O acesso ao jogo aqui descrito é uma extensão ao Modo de Jogo, e como tal implica que as pré-condições deste módulo assumem a validação prévia das condições de acesso a ele.

O jogo incluído nesta fase inicial do projeto, será um jogo de memorização e resolução de problemas simples, em que o utilizador terá um tempo limite para memorizar o melhor trajeto para chegar a um destino maximizando a sua pontuação, e um tempo limite para o traçar, como mostra a figura seguinte:

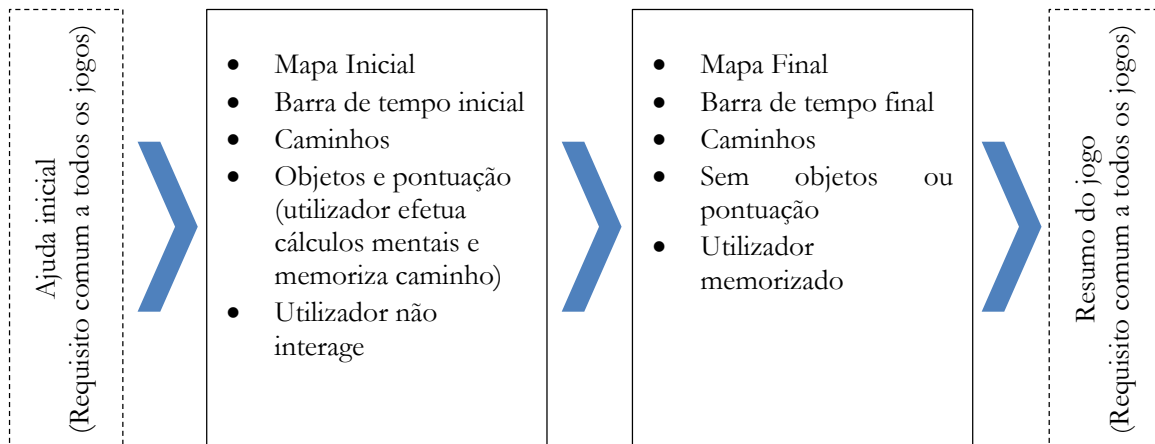


Figura 18 - Fluxo do jogo “Qual o melhor caminho”

[RF801] Mapa inicial

O jogo deverá mostrar um mapa inicial, com 1 só ponto de partida e 1 só de chegada. Este mapa de caminhos será um grafo orientado com ramificações, sem possibilidade de ciclos. Portanto, não poderá haver trajetos que permitam ao jogador escolher um caminho em que passe duas vezes pelo mesmo sítio. Ao longo destes caminhos haverão x objetos, x este que será menor ou maior consoante o grau de dificuldade. Cada objeto terá uma pontuação associada positiva ou negativa. O utilizador deverá ser informado que está a decorrer neste momento o período de memorização do mapa.

Prioridade: Muito Importante

[RF811] Barra de tempo inicial

Enquanto o mapa inicial é mostrado, deverá ser mostrada também uma barra de tempo decrescente, que indicará quanto tempo o jogador ainda tem para visualizar o mapa inicial e seus objetos. A barra de tempo a decrescer, deverá alertar para a diminuição do tempo, alterando gradualmente a sua cor de verde para vermelho.

Prioridade: Muito Importante

[RF821] Remover objetos

Depois de ultrapassado o tempo inicial, o mapa inicial ficará visível mas os objetos e sua pontuação desaparecerão, para que o jogador possa escolher o caminho memorizado. O utilizador deverá ser informado que está a decorrer o período de traçar o caminho ótimo.

Pré-condições: Tempo inicial expirado.

Prioridade: Muito Importante

[RF831] Barra de tempo final

Depois de os objetos removidos, a barra de tempo deverá encher e começar a esvaziar, indicando ao jogador que tem tempo limitado para traçar o caminho, alterando gradualmente a sua cor de verde para vermelho com o decorrer do tempo.

Pré-condições: Tempo inicial expirado.

Prioridade: Muito Importante

[RF841] Traçamento de caminho

A aplicação deverá, através do toque/clique e arrastamento do dedo/rato, detetar qual o ramo do caminho que o jogador está a indicar. O caminho deverá ser escolhido desde o ponto inicial até ao final, pelo que não poderá ser permitido seleccionar um ramo que ainda não esteja conectado ao ponto inicial, excepto se desde o último ponto seleccionado apenas houver um único caminho possível até ao ponto do toque/clique atual. Uma vez percebida qual a escolha do jogador, o ramo deverá ser logo preenchido com uma outra cor que indique ao jogador qual o caminho até agora seleccionado, sem necessidade do jogador ter que traçar o ramo completo com o dedo. Caso o jogador escolha um ramo incompatível com o caminho já traçado, este será marcado, e o caminho restante (desde a raiz ou desde a interceção destes dois caminhos) anulado.

Pré-condições: Tempo inicial expirado.

Prioridade: Muito Importante

[RF851] Alcance do ponto final

Quando o jogador alcança o ponto final, o jogo deve terminar com uma felicitação, e mostrar um botão para passar para o painel de resumo final.

Pré-condições: Tempo inicial expirado.

Prioridade: Muito Importante

[RF861] Terminus do tempo

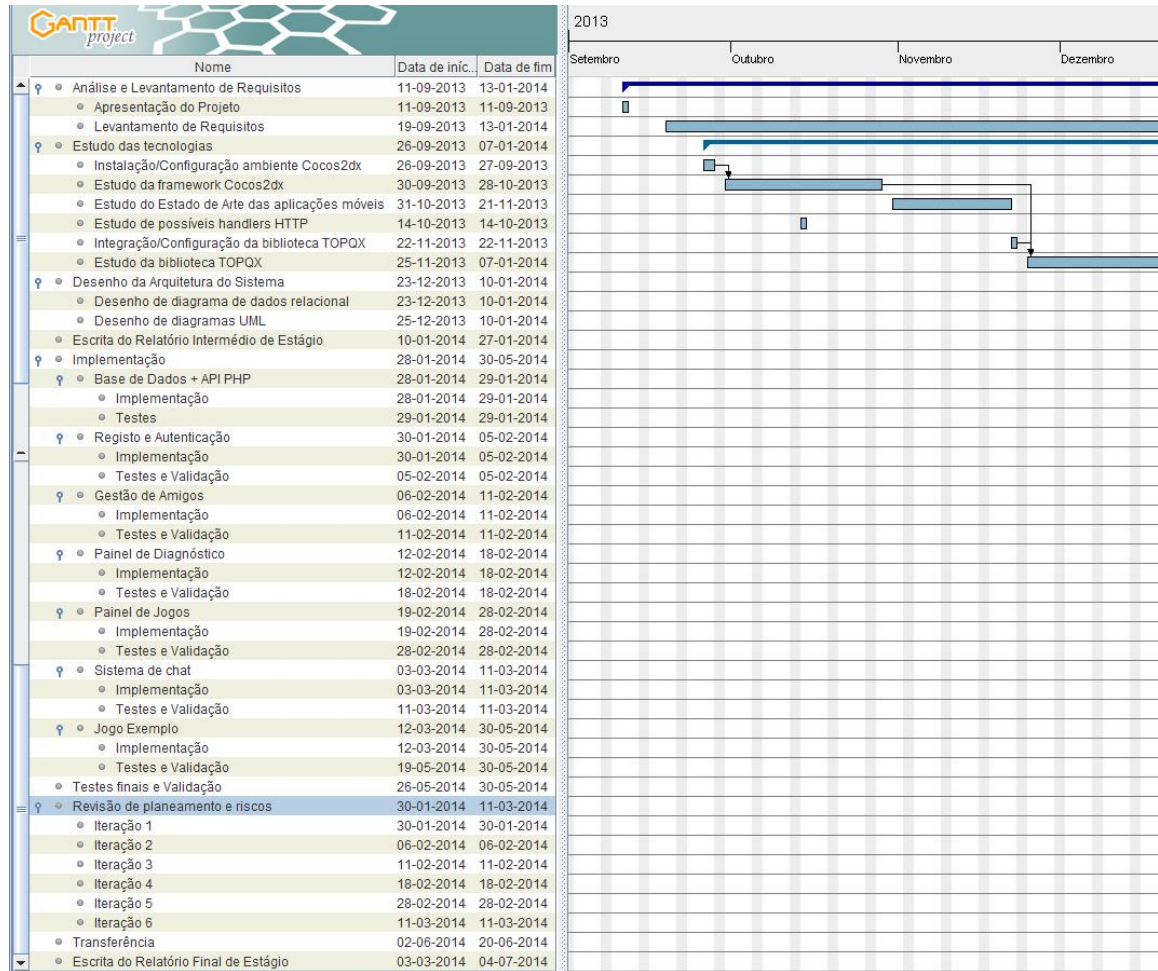
Quando o tempo estipulado para definir o caminho expira, o jogador deve ser informado com uma mensagem de lamento e questionando-o se deseja repetir, tendo para tal o jogador que pressionar um botão para o efeito, ou outro para cancelar e voltar ao painel de jogos.

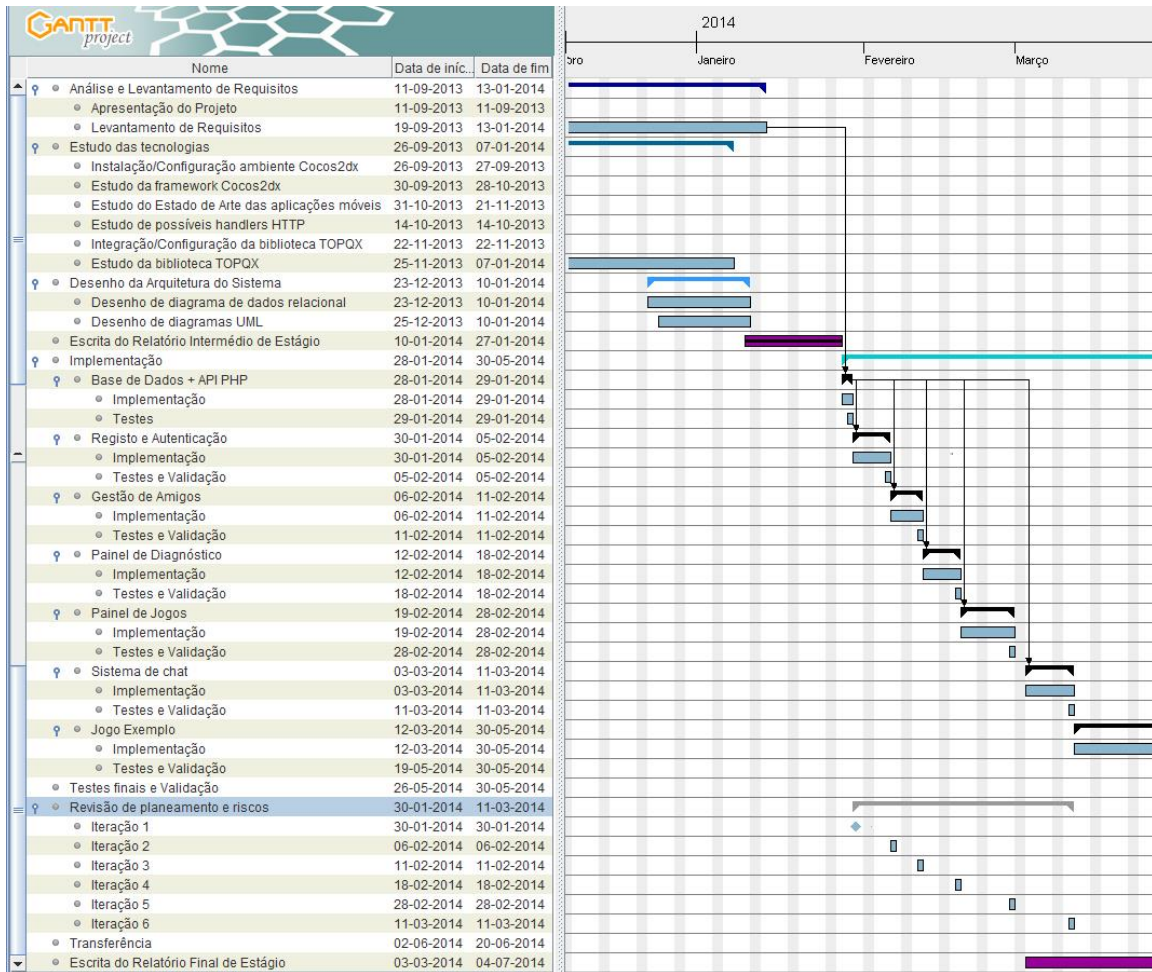
Pré-condições: Tempo inicial e final expirado.

Prioridade: Muito Importante

ANEXO B – Diagramas de Gantt da calendarização das várias tarefas

Diagrama de Gantt previsto no 1º Semestre





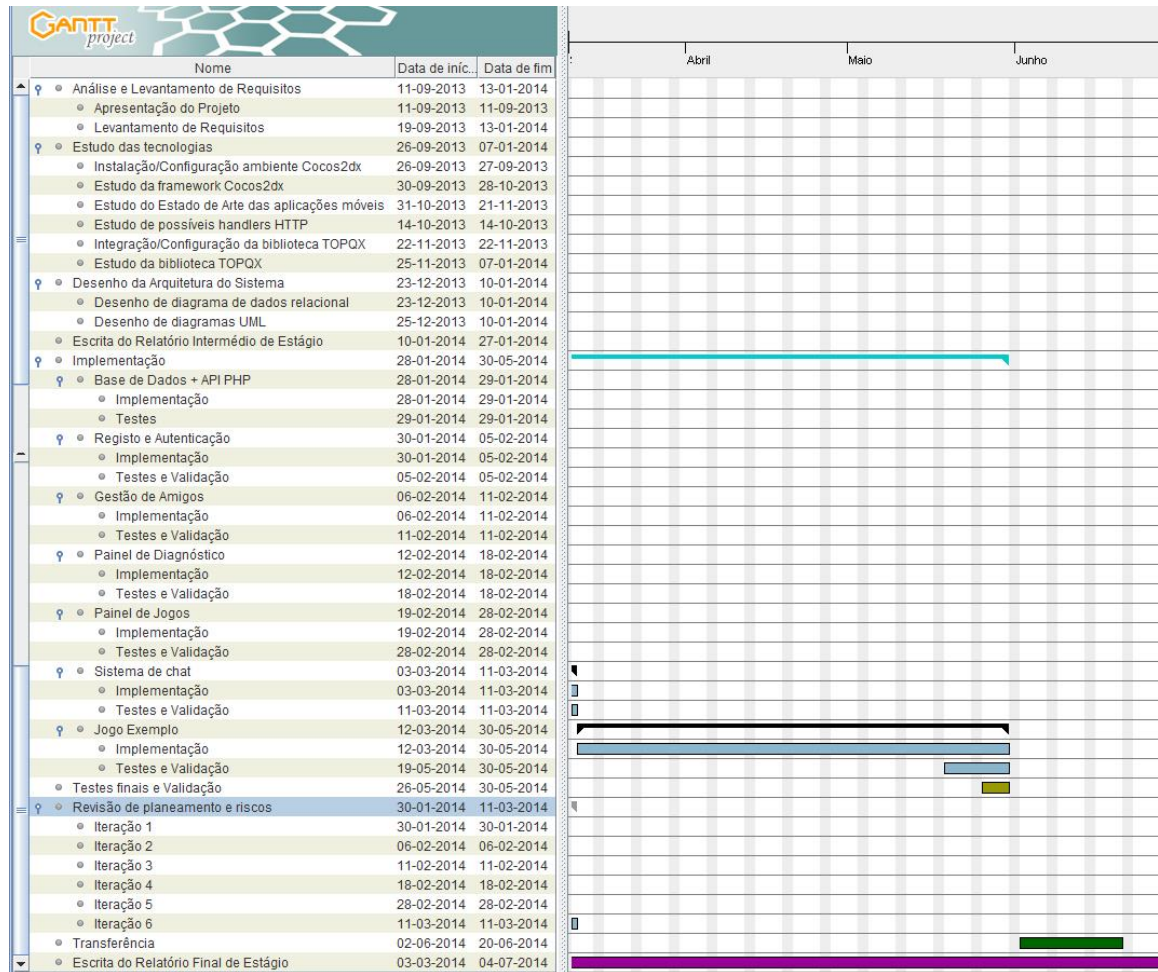
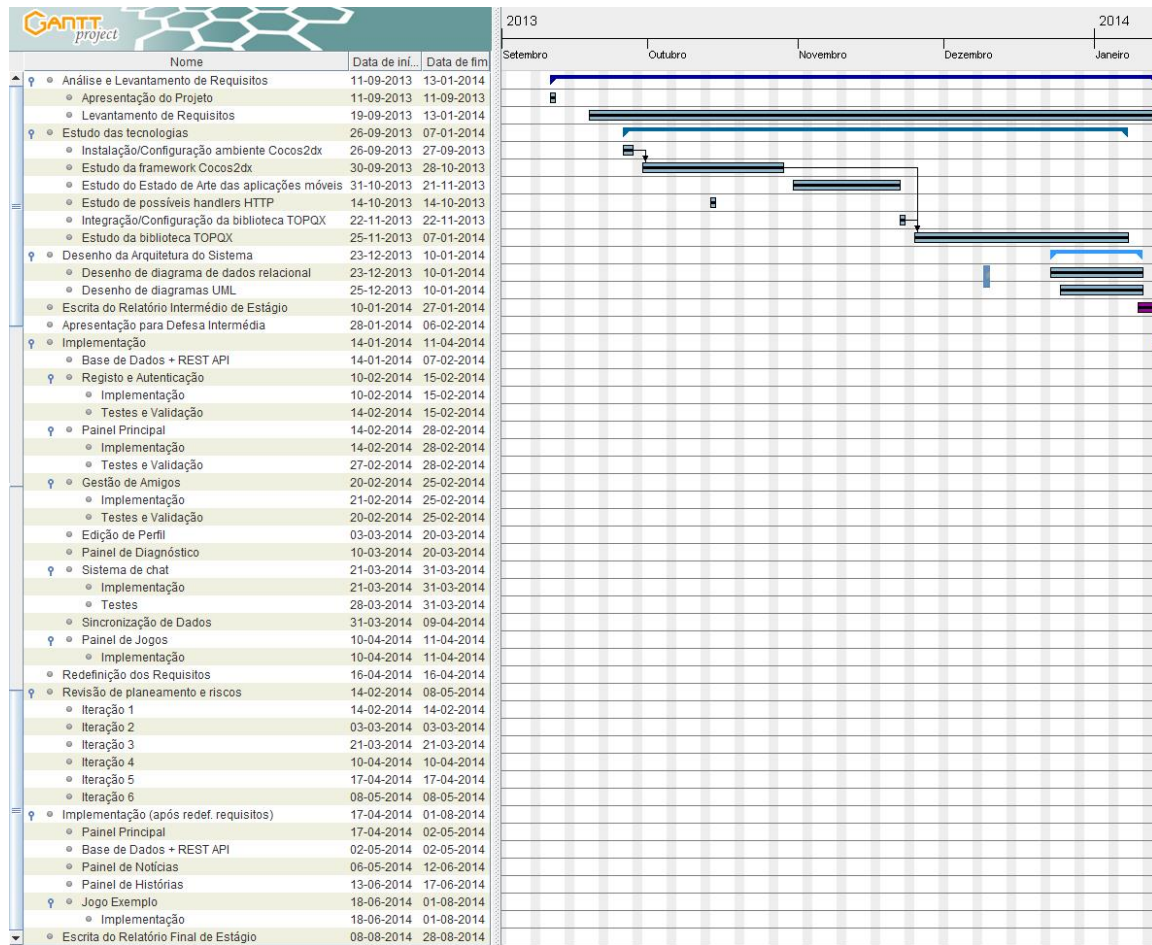
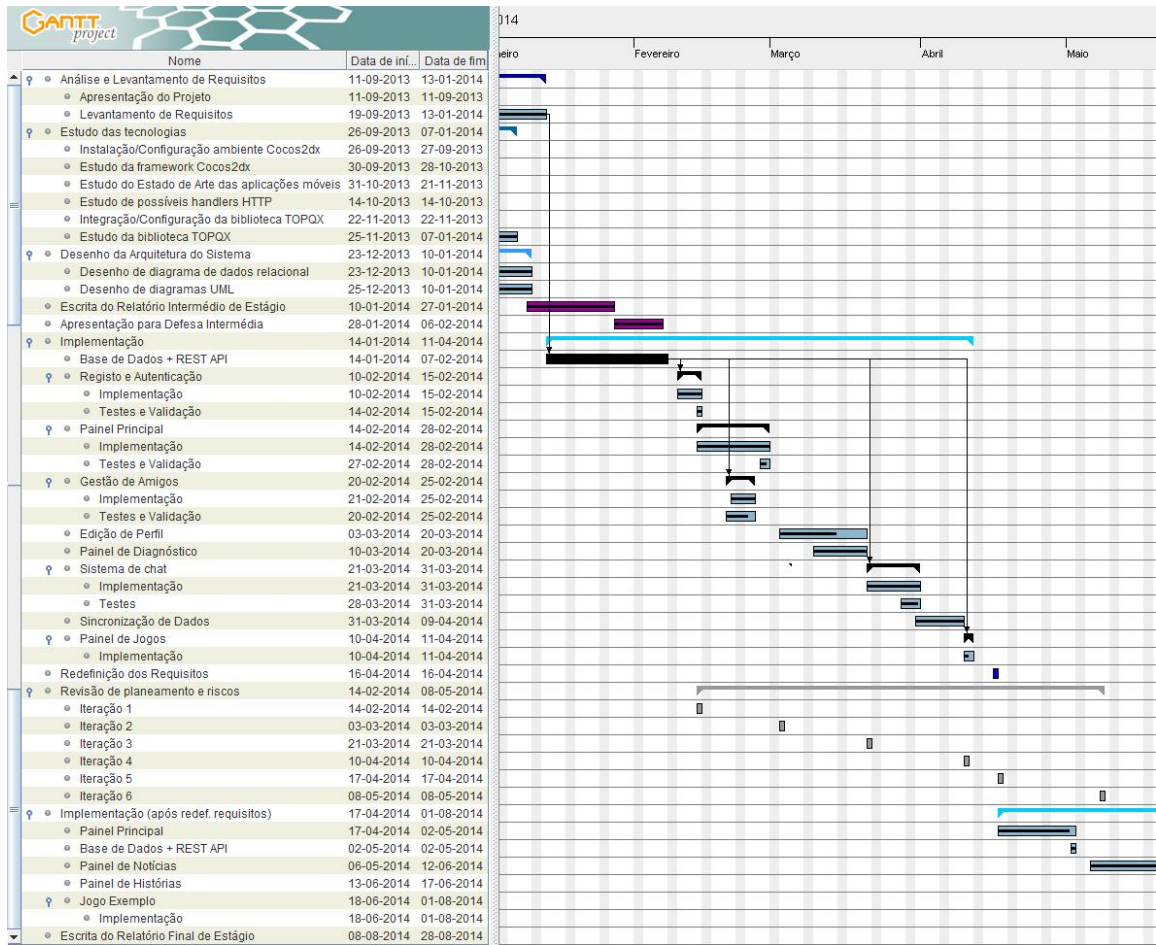
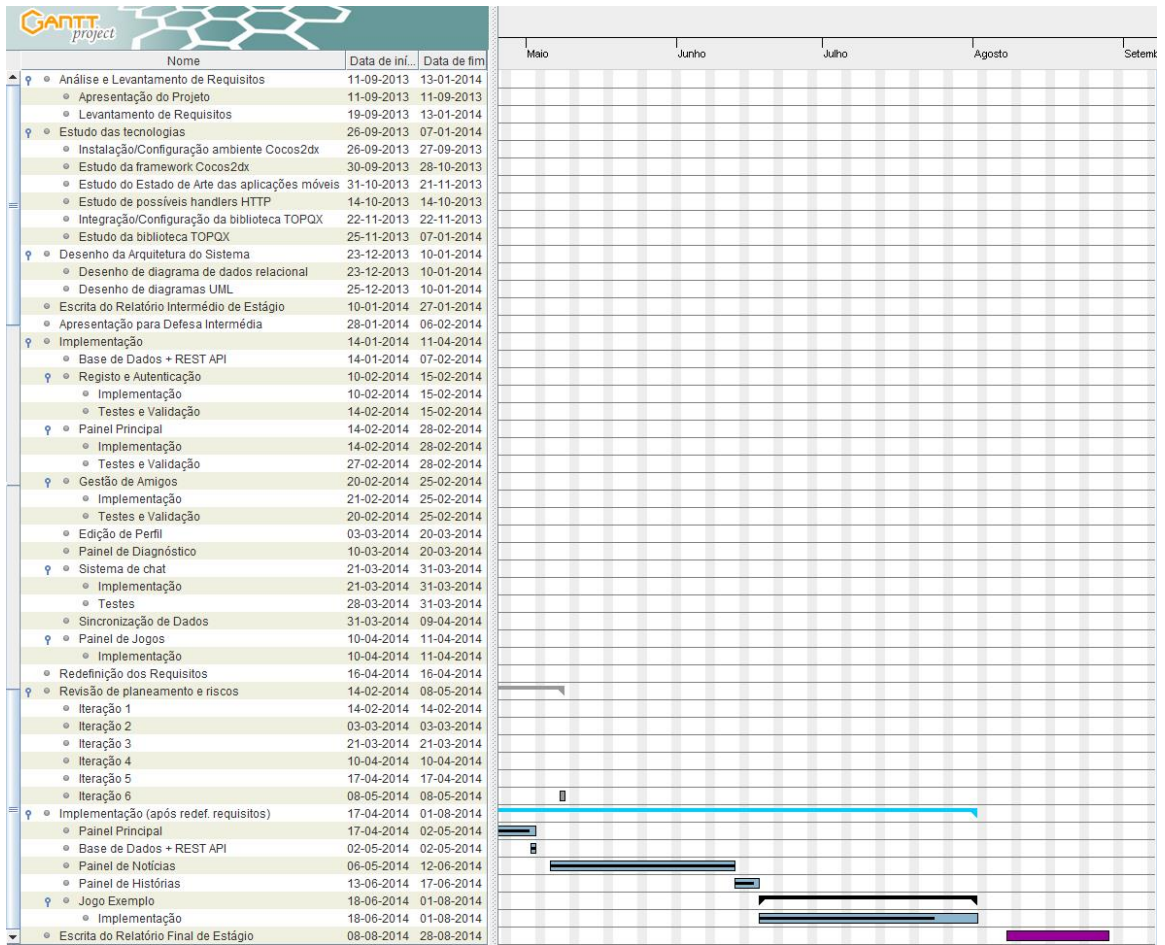


Diagrama de Gantt efetivo







ANEXO C – Exemplos de especificação da API REST

Users Handling

OPERATION	URI	METHOD	PARAMETERS			RESULTS		
			PARAMETER	MANDATORY	DESCRIPTION	RESULT	TYPE	DESCRIPTION
Get user data (including cognitive status)	/users/{user_id}	GET	user_id	yes	User ID (primary key) in DB	User Data	XML	XML with user data
						HTTP 404	HTTP Code	No user with this ID {user_id}
Get user photo	/users/{user_id}/photo	GET	user_id	yes	User ID (primary key) in DB	User Photo	JPEG or PNG	JPEG or PNG image
						HTTP 404	HTTP Code	No user with this ID {user_id}
Register new user	/users	POST	email	yes	E-mail to register	HTTP 201	HTTP Code	Successfully registered new user
			password	yes	Chooosen login password	HTTP 400	HTTP Code	Wrong or no parameters submitted
			name	yes	Name of the user	HTTP 409	HTTP Code	E-mail already registered
			fb_uid	no	Facebook user unique ID			
			birthday	no	Birthday in format yyyy/mm/dd			
			serial	no	ActiveBrain4All serial number			
			global_points	no	Global scored points			
		last_connection	no	Last loggin datetime				
Update user data	/users/{user_id}	PUT	user_id	yes	User ID (primary key) in DB	HTTP 200	HTTP Code	User data successfully updated
			email	no	E-mail of the user	HTTP 400	HTTP Code	Wrong or no parameters submitted
			password	no	Login password	HTTP 404	HTTP Code	No user with this ID {user_id}
			name	no	Name of the user	HTTP 409	HTTP Code	E-mail already registered and not owned
			fb_uid	no	Facebook user unique ID			
			birthday	no	Birthday in format yyyy/mm/dd			
		serial	no	ActiveBrain4All serial number				
Add friend	/users/{user_id}/friends	POST	user_id	yes	User ID (primary key) in DB	HTTP 201	HTTP Code	Friend added successfully
			friendid	yes	Friend ID (primary key) in DB	HTTP 409	HTTP Code	Friendship already exists
Remove friend	/users/{user_id}/friends/{friend_id}	DELETE	user_id	yes	User ID (primary key) in DB	HTTP 200	HTTP Code	Friend removed successfully
			friend_id	yes	Friend ID (primary key) in DB	HTTP 404	HTTP Code	No user with this ID {user_id} or friend with {friend_id}
Get user stories	/users/{user_id}/stories	GET	user_id	yes	User ID (primary key) in DB	User Stories	XML	XML with stories subject and body
						HTTP 404	HTTP Code	No user with this ID {user_id}

Communication Handling

OPERATION	URI	METHOD	PARAMETERS			RESULTS		
			PARAMETER	MANDATORY	DESCRIPTION	RESULT	TYPE	DESCRIPTION
Get user new messages	/users/{user_id}/inbox/	GET	user_id	yes	User ID (primary key) in DB	New Messages	XML	XML with new messages (from, to, time, text)
						HTTP 304	HTTP Code	No new messages
						HTTP 404	HTTP Code	No user with this ID {user_id}
Send message to user	/users/{user_id}/inbox/	POST	user_id	yes	Destinatary user ID (primary key) in DB	HTTP 201	HTTP Code	Message sent successfully
			id	yes	Sender user ID (primary key) in DB	HTTP 400	HTTP Code	Wrong or no parameters submitted
			text	yes	Message body to send	HTTP 404	HTTP Code	No user with this ID {user_id}
Check for game invitation	/users/{user_id}/invitations	GET	user_id	yes	User ID (primary key) in DB	Invitations	XML	XML with new invitations (from, to, time, text)
						HTTP 304	HTTP Code	No new invitations
						HTTP 404	HTTP Code	No user with this ID {user_id}
Reject game invitation	/users/{user_id}/invitations/{id}	DELETE	user_id	yes	Destinatary user ID (primary key) in DB	HTTP 200	HTTP Code	Invitation rejected successfully
			id	yes	Invitation ID (primary key) in DB	HTTP 400	HTTP Code	Wrong user_id and id combination
						HTTP 404	HTTP Code	No user with this ID {user_id} and/or no game with this ID {id}

News Handling

OPERATION	URI	METHOD	PARAMETERS			RESULT	RESULTS	
			PARAMETER	MANDATORY	DESCRIPTION		TYPE	DESCRIPTION
Get last news	/news	GET				Last News	XML	XML with last news title and body

Communication Handling

OPERATION	URI	METHOD	PARAMETERS			RESULT	RESULTS	
			PARAMETER	MANDATORY	DESCRIPTION		TYPE	DESCRIPTION
Get user new messages	/users/{user_id}/inbox/	GET	user_id	yes	User ID (primary key) in DB	New Messages	XML	XML with new messages (from, to, time, text)
						HTTP 304	HTTP Code	No new messages
						HTTP 404	HTTP Code	No user with this ID {user_id}
Send message to user	/users/{user_id}/inbox/	POST	user_id	yes	Destinatary user ID (primary key) in DB	HTTP 201	HTTP Code	Message sent successfully
			id	yes	Sender user ID (primary key) in DB	HTTP 400	HTTP Code	Wrong or no parameters submitted
			text	yes	Message body to send	HTTP 404	HTTP Code	No user with this ID {user_id}
Check for game invitation	/users/{user_id}/invitations	GET	user_id	yes	User ID (primary key) in DB	Invitations	XML	XML with new invitations (from, to, time, text)
						HTTP 304	HTTP Code	No new invitations
						HTTP 404	HTTP Code	No user with this ID {user_id}
Reject game invitation	/users/{user_id}/invitations/{id}	DELETE	user_id	yes	Destinatary user ID (primary key) in DB	HTTP 200	HTTP Code	Invitation rejected successfully
			id	yes	Invitation ID (primary key) in DB	HTTP 400	HTTP Code	Wrong user_id and id combination
						HTTP 404	HTTP Code	No user with this ID {user_id} and/or no game with this ID {id}

ANEXO D – Exemplos de interfaces implementadas

Login Interface



Fig. D1 – Interface de Login por seleção de utilizador frequente

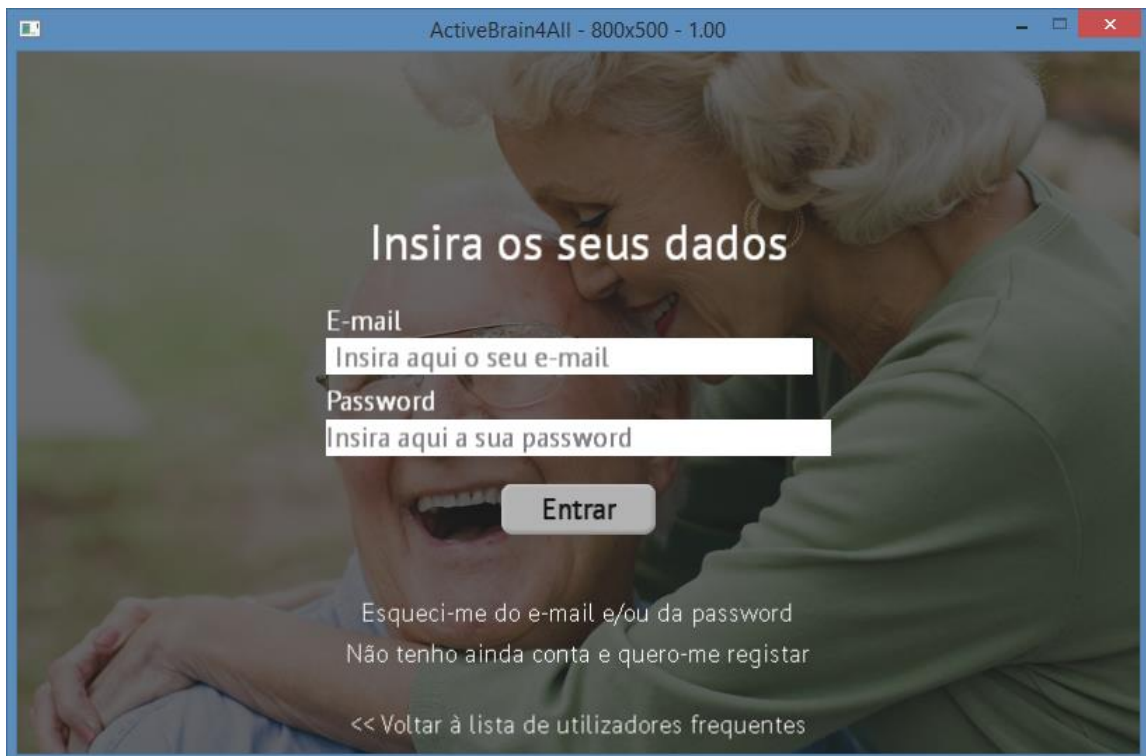


Fig. D2 – Interface de Login por introdução de e-mail e password

Menu Principal



Fig. D3 – Interface do Menu Principal com escolha de uma das 3 componentes

Perfil pessoal



Fig. D4 – Interface de gestão de dados, com visualização da evolução do estado cognitivo

Notícias



Fig. D5 – Interface de visualização de notícias

Histórias



Fig. D6 – Interface de visualização e gestão de histórias

Jogo

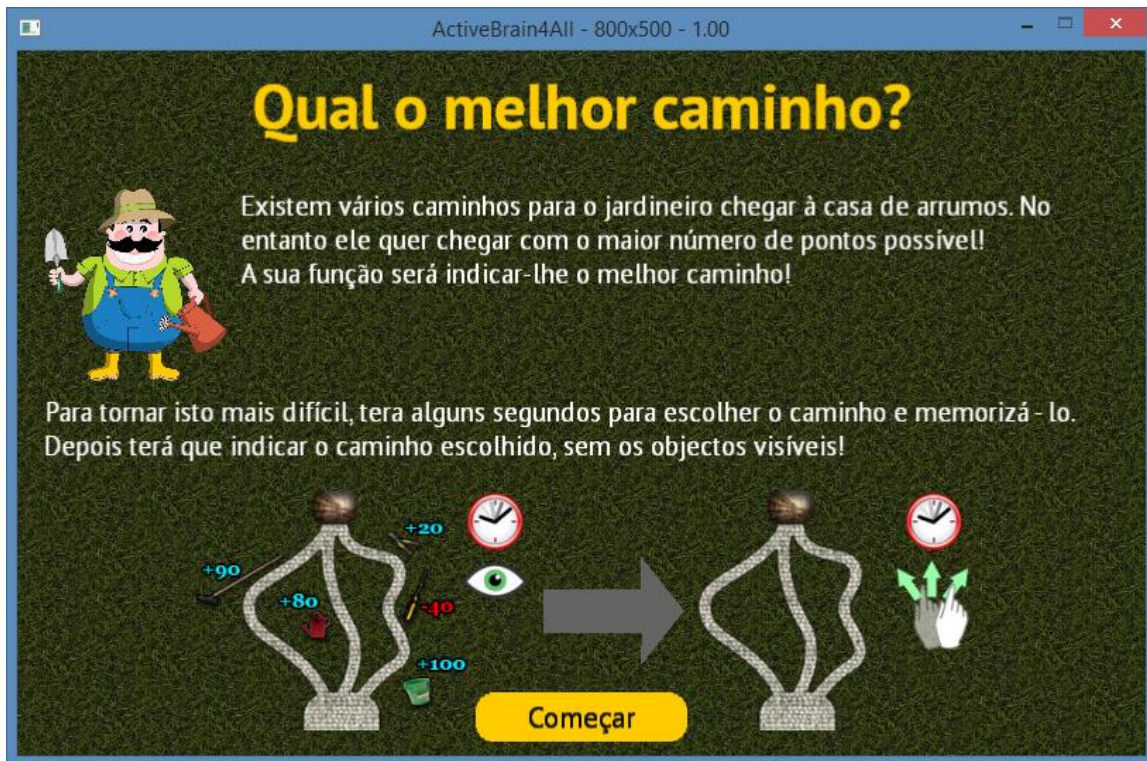


Fig. D7 – Ajuda inicial do jogo “Qual o melhor caminho”

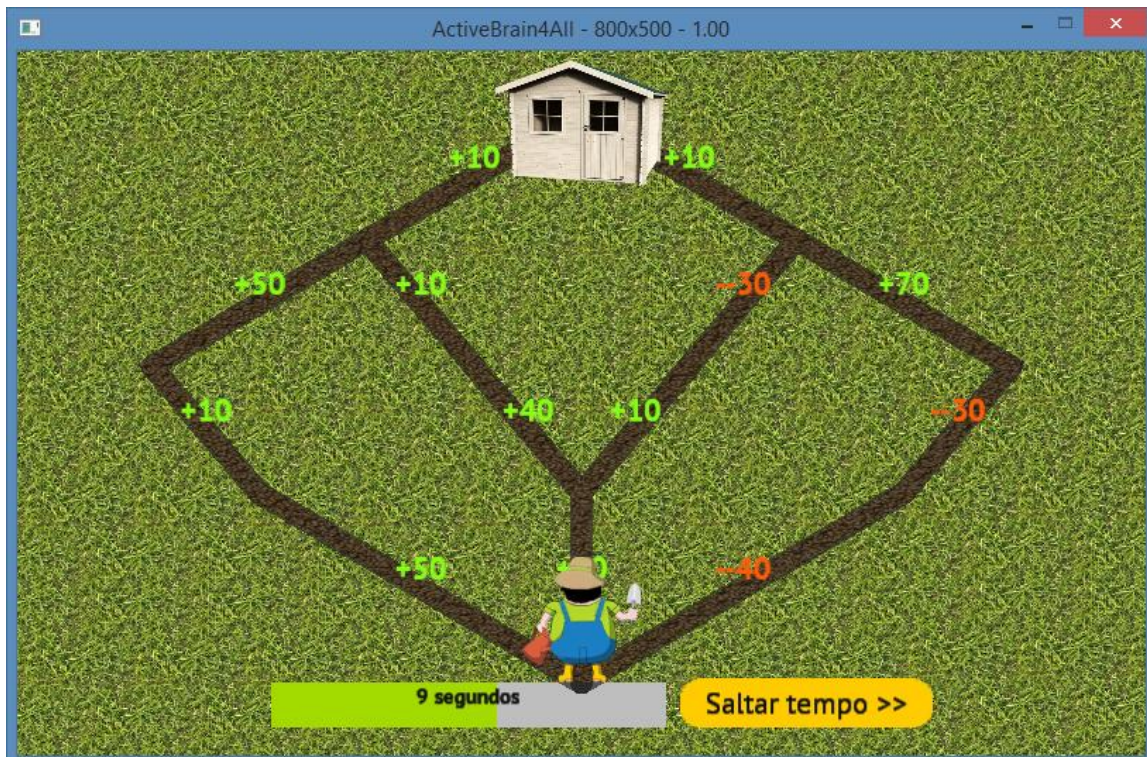


Fig. D8 – Decorrer do jogo, nomeadamente o período de cálculo e memorização do melhor caminho

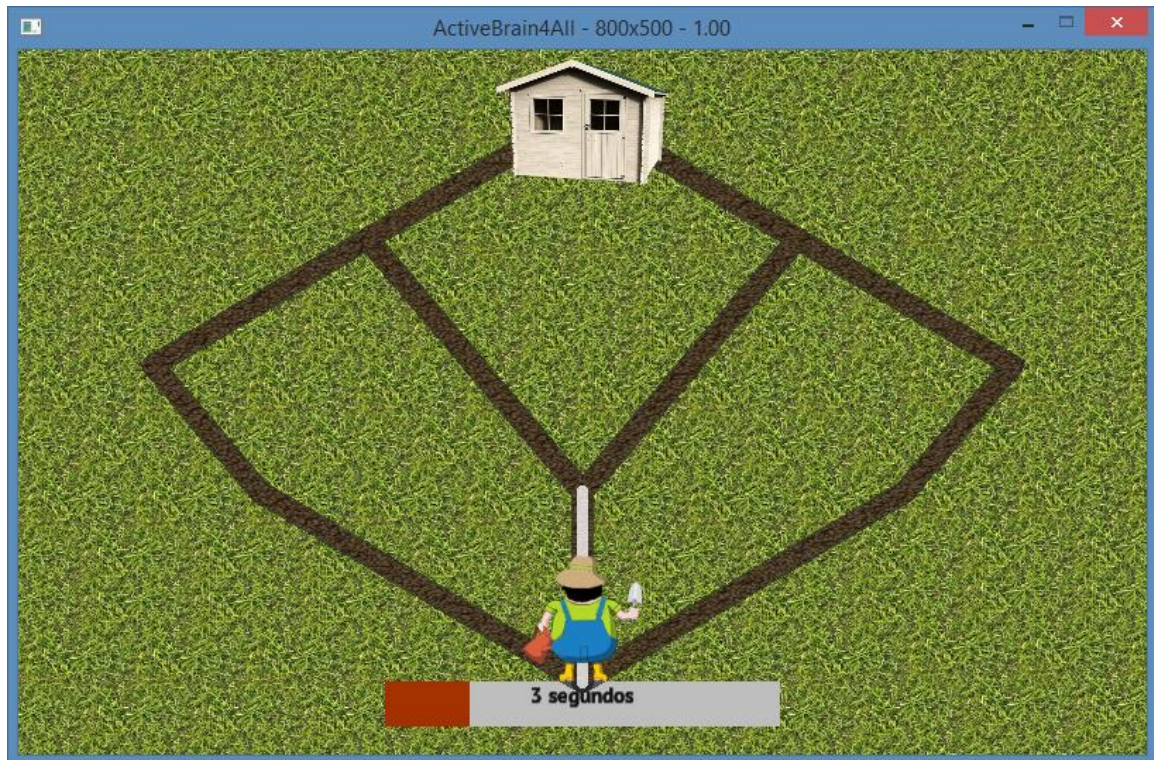


Fig. D9 – Decorrer do jogo. Período de traçamento do melhor caminho antes memorizado



Fig. D10 – Painel de final de jogo, mostrando pontuação e progresso cognitivo