**LABDSOFT**

**ISEP – Mestrado em Engenharia Informática**

**Documento Arquitetura de Software**

**Banco de Gâmetas - GAM**

Trabalho prático nr.2 – iteração 1

**Histórico de Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Autores | Descrição | Data |
| **1.0** | **Grupo 1**  Ana Barros  Daniel Bento  Manuel Correia  Maria Almeida  Tiago Gonçalves |  | **2017-10-22** |
| **1.2** | **Grupo 1**  Ana Barros  Daniel Bento  Manuel Correia  Maria Almeida  Tiago Gonçalves |  | **2017-11-26** |
| **1.3** | **Grupo 1**  Ana Barros  Daniel Bento  Manuel Correia  Maria Almeida  Tiago Gonçalves |  | **2017-12-10** |

2017

**Índice**

[**LABDSOFT** 0](#_Toc500429765)

[**ISEP – Mestrado em Engenharia Informática** 0](#_Toc500429766)

[**Banco de Gâmetas - GAM** 0](#_Toc500429767)

[Trabalho prático nr.2 – iteração 1 0](#_Toc500429768)

[**Histórico de Revisão** 0](#_Toc500429769)

[**Índice** 2](#_Toc500429770)

[**Índice de Ilustrações** 5](#_Toc500429771)

[**Índice de Tabelas** 6](#_Toc500429772)

[1 Introdução 7](#_Toc500429773)

[1.1 Propósito 7](#_Toc500429774)

[1.2 Scope do produto 7](#_Toc500429775)

[1.3 Funcionalidades propostas 7](#_Toc500429776)

[2 Decisões tomadas 9](#_Toc500429777)

[3 Artefactos 10](#_Toc500429778)

[3.1 Diagrama de contexto 10](#_Toc500429779)

[3.2 Modelo de dados 11](#_Toc500429780)

[3.3 Diagrama de arquitetura 12](#_Toc500429781)

[5 Funcionalidades implementadas 13](#_Toc500429782)

[5.1 REQ 1 – Registo inicial do dador 13](#_Toc500429786)

[5.1.1 Análise de requisitos 13](#_Toc500429787)

[5.1.2 Diagrama de sequência 14](#_Toc500429788)

[5.1.3 Ilustração da funcionalidade 15](#_Toc500429789)

[5.2 REQ 2 – Validação dos dados do dador 16](#_Toc500429790)

[5.2.1 Análise de requisitos 16](#_Toc500429791)

[5.2.2 Diagrama de sequência 17](#_Toc500429792)

[5.2.1 Ilustração da funcionalidade 18](#_Toc500429793)

[5.3 REQ 3 – Gestão de perfis de utilizadores 19](#_Toc500429794)

[5.3.1 Análise de requisitos 19](#_Toc500429795)

[5.3.2 Diagrama de sequência 20](#_Toc500429796)

[5.3.3 Ilustração da funcionalidade 21](#_Toc500429797)

[5.4 REQ 4 – Configuração do template de perguntas médicas 22](#_Toc500429798)

[5.4.1 Análise de requisitos 22](#_Toc500429799)

[5.4.2 Diagrama de sequência 23](#_Toc500429800)

[5.4.3 Ilustração da funcionalidade 24](#_Toc500429801)

[5.5 REQ 5 – Consultas médicas via inquérito template 25](#_Toc500429802)

[5.5.1 Análise de requisitos 25](#_Toc500429803)

[5.5.2 Diagrama de sequência 26](#_Toc500429804)

[5.5.3 Ilustração da funcionalidade 27](#_Toc500429805)

[5.6 REQ 6 – Registo de amostras 28](#_Toc500429806)

[5.6.1 Análise de requisitos 28](#_Toc500429807)

[5.6.2 Diagrama de sequência 29](#_Toc500429808)

[5.6.3 Ilustração da funcionalidade 30](#_Toc500429809)

[5.7 REQ 7 – Pedido das análises de sangue iniciais 32](#_Toc500429810)

[5.7.1 Análise de requisitos 32](#_Toc500429811)

[5.7.2 Diagrama de sequência 33](#_Toc500429812)

[5.7.3 Ilustração da funcionalidade 34](#_Toc500429813)

[5.8 REQ 8 – Ciclo da dádiva (abertura e consulta de informação) 35](#_Toc500429814)

[5.8.1 Análise de requisitos 35](#_Toc500429815)

[5.8.2 Diagrama de sequência 36](#_Toc500429816)

[5.8.3 Ilustração da funcionalidade 37](#_Toc500429817)

[5.9 REQ 9 – Marcação de consultas 38](#_Toc500429818)

[5.9.1 Análise de requisitos 38](#_Toc500429819)

[5.9.2 Diagrama de sequência 39](#_Toc500429820)

[5.9.3 Ilustração da funcionalidade 40](#_Toc500429821)

[5.10 REQ 10 – Aprovação/Rejeição do ciclo da dádiva 41](#_Toc500429822)

[5.10.1 Análise de requisitos 41](#_Toc500429823)

[5.10.2 Diagrama de sequência 42](#_Toc500429824)

[5.10.3 Ilustração da funcionalidade 43](#_Toc500429825)

[5.11 REQ 11 – Lista de trabalhos do laboratório 44](#_Toc500429826)

[5.11.1 Análise de requisitos 44](#_Toc500429827)

[5.11.2 Diagrama de sequência 44](#_Toc500429828)

[5.11.3 Ilustração da funcionalidade 45](#_Toc500429829)

[5.12 REQ 12 – Registo de espermograma 46](#_Toc500429830)

[5.12.1 Análise de requisitos 46](#_Toc500429831)

[5.12.2 Diagrama de sequência 47](#_Toc500429832)

[5.12.3 Ilustração da funcionalidade 48](#_Toc500429833)

[5.13 REQ 13 – Validação do espermograma 50](#_Toc500429834)

[5.13.1 Análise de requisitos 50](#_Toc500429835)

[5.13.2 Diagrama de sequência 51](#_Toc500429836)

[5.13.3 Ilustração da funcionalidade 52](#_Toc500429837)

[5.14 REQ 14 – Criopreservação da amostra 53](#_Toc500429838)

[5.14.1 Análise de requisitos 53](#_Toc500429839)

[5.14.2 Diagrama de sequência 54](#_Toc500429840)

[5.14.3 Ilustração da funcionalidade 55](#_Toc500429841)

[5.15 REQ 15 – Visualização da alocação das amostras no Banco de criopreservação 57](#_Toc500429842)

[5.15.1 Análise de requisitos 57](#_Toc500429843)

[5.15.2 Diagrama de sequência 58](#_Toc500429844)

[5.15.3 Ilustração da funcionalidade 58](#_Toc500429845)

[5.16 REQ 17 – Registo de pedidos de gâmetas 59](#_Toc500429846)

[5.16.1 Análise de requisitos 59](#_Toc500429847)

[5.16.2 Diagrama de sequência 60](#_Toc500429848)

[5.16.3 Ilustração da funcionalidade 61](#_Toc500429849)

[5.17 REQ 18 – Registo de materiais usados (sem integração) 63](#_Toc500429850)

[5.17.1 Análise de requisitos 63](#_Toc500429851)

[5.17.2 Diagrama de sequência 64](#_Toc500429852)

[5.17.3 Ilustração da funcionalidade 65](#_Toc500429853)

[4 Funcionalidades não implementadas 65](#_Toc500429854)

[5 Pipeline 66](#_Toc500429855)

[6 Distribuição de tarefas 68](#_Toc500429856)

[7 Tempos 69](#_Toc500429857)

[8 Medidas de qualidade de software 70](#_Toc500429858)

**Índice de Ilustrações**

[Figura 1 - Diagrama de Contexto 10](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477199)

[Figura 2 - Modelo de dados 11](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477200)

[Figura 3 - Diagrama de arquitetura 12](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477201)

[Figura 4 - Excerto do ficheiro de resultados (testes) 66](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477202)

[Figura 5 – Artefactos criados após stage Archive 67](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477203)

[Figura 6 - Stage View Pipeline 67](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477204)

[Figura 7 - REQ\_1: Página Inicial 15](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477205)

[Figura 8 - REQ\_1: View de Registo do Dador 15](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477206)

[Figura 9 - REQ\_2: View Lista de Dadores 18](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477207)

[Figura 10 - REQ\_2: View da Validação de Dador 18](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477208)

[Figura 11 - REQ\_3: View da Lista de Perfis 21](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477209)

[Figura 12 - REQ\_3: View da Criação de um novo perfil 21](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477210)

[Figura 13: REQ\_3: View de edição de um perfil 21](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477211)

[Figura 14 - REQ\_4: View de edição de um questionário 24](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477212)

[Figura 15 - REQ\_5: View da lista de possíveis dadores 27](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477213)

[Figura 16 - REQ5: View dos detalhes do dador 27](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477214)

[Figura 17: REQ\_5: View do preenchimento do formulário 27](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477215)

[Figura 18 - REQ\_6: View da lista de amostras 30](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477216)

[Figura 19 - REQ\_6: View de criação de uma amostra 30](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477217)

[Figura 20 - REQ\_6 - View de detalhes de uma amostra 31](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477218)

[Figura 21 - REQ\_6: View de edição de uma amostra 31](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477219)

[Figura 22 - REQ\_7: View da listagem de dadores com amostras pendentes de análise 34](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477220)

[Figura 23 - REQ\_7: View dos resultados das análises gerados pela API iAnalysis 34](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477221)

[Figura 24 - REQ\_8: View dos processos pendentes e abertos 37](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477222)

[Figura 25 - REQ\_8: View de abertura do processo 37](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477223)

[Figura 26 - REQ\_9: View da lista de consultas 40](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477224)

[Figura 27 - REQ\_9: View da marcação de uma consulta 40](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477225)

[Figura 28 - REQ\_10: View da lista de processos pendentes de aprovação 43](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477226)

[Figura 29 - REQ\_10: View da aprovação/rejeição de um processo 43](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477227)

[Figura 30 - REQ\_11: View da lista de amostras por analisar 45](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477228)

[Figura 31 -REQ\_11: View da lista de amostras analisadas 45](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477229)

[Figura 32 - REQ\_12 – View da lista de espermogramas 48](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477230)

[Figura 33 - REQ\_12: View de criação de espermograma 48](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477231)

[Figura 34 - REQ\_12: View dos detalhes do espermograma 49](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477232)

[Figura 35 - REQ\_13: View da lista de espermogramas 52](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477233)

[Figura 36 - REQ\_13: View de aprovação/rejeição de um espermograma 52](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477234)

[Figura 37 - REQ\_14: View da lista de amostras analisadas 55](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477235)

[Figura 38 - REQ\_14: View dos detalhes de uma amostra 55](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477236)

[Figura 39 - REQ\_14: View da criopreservação de uma amostra 56](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477237)

[Figura 40 - REQ\_15: View da lista de alocação das amostras 58](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477238)

[Figura 41 - REQ\_17: View da lista de pedidos de gâmetas 61](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477239)

[Figura 42 - REQ\_17: View de detalhes de um pedido de gâmetas 61](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477240)

[Figura 43 - REQ\_17: View de edição de um pedido de gâmetas 62](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477241)

[Figura 44 - REQ\_17: View da criação de um pedido de gâmetas 62](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477242)

[Figura 45 - REQ\_18: View da lista de materiais usados 65](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477243)

[Figura 46 - REQ\_18: View do registo de material utilizado 65](file:///C:\Users\mcorreia\Documents\GitKrakenProjects\labdsoft-2017-g1\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499477244)

**Índice de Tabelas**

Tabela 1 - Funcionalidades propostas 8

Tabela 2 – Distribuição tarefas 68

Tabela 3 - Duração de cada requisito 69

# Introdução

# Propósito

Este documento foi desenvolvido no contexto da segunda iteração do trabalho prático da disciplina de Laboratório Desenvolvimento Software (LABDSOFT), integrada no Mestrado de Engenharia Informática (MEI) do Instituto Superior de Engenharia do Porto (ISEP). Tem como propósito descrever os processos inerentes à implementação de alguns requisitos definidos para a aplicação de software - iGAM. No que diz respeito às funcionalidades a implementar, iremos garantir que o processo consegue fluir com todos os intervenientes, desde a sua inscrição inicial até à sua aprovação ou rejeição.

Ao longo do documento serão abordados tópicos relacionados com algumas das decisões tomadas pelo grupo, artefactos relevantes a nível arquitetural e tecnológico incluindo a especificação do pipeline criado. Para além disso, serão apresentadas as funcionalidades implementadas bem como a distribuição de tarefas entre o grupo de trabalho.

## Scope do produto

A GAM pretende uma solução de software que consiga monitorizar os dois grandes processos que neste momento existem apenas em formato papel. A gestão do conjunto de etapas que constituem os processos é outro grande foco.

O primeiro processo denomina-se por “**Ciclo de Dádiva**”, é muito rigoroso e constitui várias etapas. Existem dadores que realizam amostras no banco. Após verificação e validação de critérios bem definidos, o dador é considerado um dador “válido”. As amostras de espermatozoides ficam crio preservadas para mais tarde puderem ser usadas por casais inférteis.

O segundo processo denomina-se por “**Resposta a Pedidos de Gâmetas**”. Existem clínicas/hospitais que nas suas consultas de procriação recomendam técnicas de fertilidade como por exemplo, a fertilização in vitro com dadores anónimos. Estas clínicas solicitam amostras para puderem ser usadas nestes casais. O pedido é feito ao banco (formato papel), a respetiva clínica envia o conjunto de caraterísticas do casal e o banco é responsável por selecionar o dador que melhor obedece aos requisitos.

Em suma, o objetivo principal desta solução é automatizar estes dois processos, mas também ser capaz de apresentar ao cliente dados estatísticos relativos a ambos os processos.

## Funcionalidades propostas

Para esta iteração, temos um total de dezoito requisitos que pretendemos implementar no sistema de software iGAM e dez atores envolvidos (dador, informático, enfermeira coordenadora, médico, enfermeiro, embriologista, diretora laboratório, embriologista, laboratório sangue (externo), Clínica PMA) – Tabela 1.

Tabela 1 - Funcionalidades propostas

|  |  |  |
| --- | --- | --- |
| Id | Identificação | Ator |
| REQ\_1 | Registo inicial do dador (sem fotografia) | Dador |
| REQ\_2 | Validação dos dados do dador | Enfermeira coordenadora |
| REQ\_3 | Gestão de perfis de utilizadores | Informático |
| REQ\_4 | Configuração do template de perguntas médicas | Médico |
| REQ\_5 | Consultas médicas via inquérito template | Médico |
| REQ\_6 | Registo de amostras | Enfermeira |
| REQ\_7 | Pedido das análises de sangue iniciais | Médico |
| REQ\_8 | Ciclo da dádiva (abertura e consulta de informação) | Enfermeira coordenadora |
| REQ\_9 | Marcação de consultas | Enfermeira |
| REQ\_10 | Aprovação / Rejeição do ciclo da dádiva | Médico |
| REQ\_11 | Lista de trabalho do laboratório | Embriologista |
| REQ\_12 | Registo de espermograma | Embriologista |
| REQ\_13 | Validação do espermograma | Diretora laboratório |
| REQ\_14 | Criopreservação da amostra | Embriologista |
| REQ\_15 | Visualização da alocação das amostras no Banco de criopreservação | Embriologista |
| REQ\_16 | Integração com o laboratório de análises de sangue | Lab. Sangue (externo) |
| REQ\_17 | Registo de pedidos de gâmetas | Clínica PMA |
| REQ\_18 | Registo de materiais usados (sem integração) | Embriologista |

# Decisões tomadas

A nível de organização da equipa de desenvolvimento, optamos por utilizar a aplicação Slack como veículo de comunicação entre todos os elementos. O uso desta plataforma de comunicação permitiu-nos fazer chamadas de grupo, debater problemas, partilhar excertos de código e manter-nos informados sobre o estado de cada *issue*, uma vez que foi adicionado um *plugin* de integração com o Bitbucket. Importa também referir que no Bitbucket, foi criado um *board* associado ao Trello, onde foram sendo adicionadas as tarefas a desenvolver ao longo desta iteração, organizadas por etapas. Deste modo, todos os elementos da equipa de desenvolvimento sabiam o estado de cada *issue*.

Quanto ao desenvolvimento do produto, optamos por utilizar a linguagem de programação ASP.NET Core, uma vez que todos os elementos do grupo estavam familiarizados com a mesma. De realçar que a decisão recaiu sobre ASP.NET Core e não sobre ASP.NET pelo fato da primeira ser multiplataforma e modular, o que significa que as aplicações são mais leves e com melhor desempenho. Já a nível arquitetural, decidimos seguir uma arquitetura MVC (Model View Controller) porque, mais uma vez, os elementos do grupo já estavam familiarizados com a mesma e consideramos que nos dá garantias de manter o código organizado, estruturado e com uma adequada separação de responsabilidades. Uma das vantagens deste tipo de arquitetura é o fato de permitir a diversos elementos trabalhar sobre o mesmo repositório, sem que se verifiquem conflitos na junção de vários pedaços de código.

Por sua vez, quanto à persistência de dados, numa primeira instância recorremos a uma base de dados local providenciada pelo ambiente de desenvolvimento usado (Visual Studio 2017), onde implementamos um *script* para povoar a base de dados, de cada máquina onde corresse a solução, com alguns dados. Porém, esta solução não nos permitia apresentar uma aplicação escalável pelo que tomamos a decisão de migrar os dados para a *cloud*. Deste modo, temos uma única base de dados, partilhada e acessível por todos os elementos da equipa de desenvolvimento.

Por fim, como ferramenta de automação de todo o processo foi utilizada a plataforma Jenkins. O processo foi então dividido em cinco etapas, desde a conexão ao projeto alojado no Bitbucket, passando pela *build* do mesmo e a execução de alguns testes unitários, e terminando na publicação dos resultados dos testes e na geração de um zip com todos os dados.

# Artefactos

## Diagrama de contexto

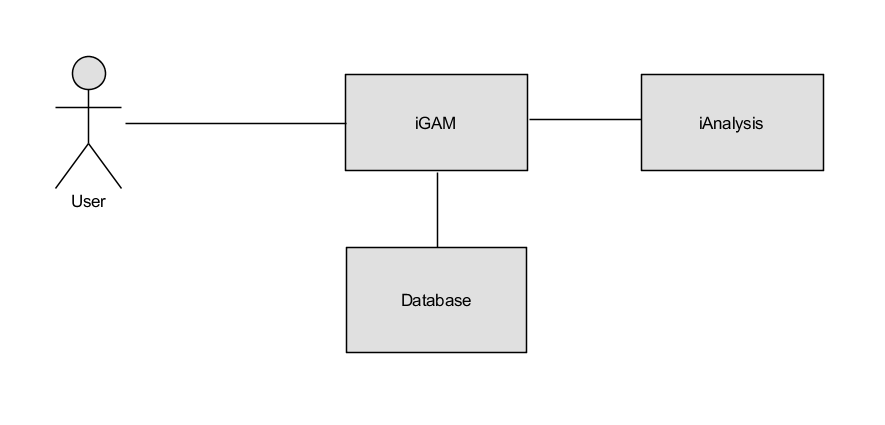


Figura 1 - Diagrama de Contexto

Na Figura 1 é apresentado o Diagrama de Contexto da aplicação, onde podemos ver que os utilizadores (Dador, Médico, Enfermeiro, Embriologista, Diretor Geral, Clinicas PMA, Informático e Assistente Social) vão interagir com a aplicação *web* iGAM. Esta aplicação pode ser executada a partir de qualquer navegador de Internet e, com exceção dos Dadores, para usufruir das suas funcionalidades é necessário ser um utilizador registado no sistema. Por sua vez, a plataforma iGAM vai interagir com uma base de dados alojada na *cloud*, para garantir a persistência de dados. Por fim, a plataforma iGAM recorre a uma *web* API, denominada iAnalysis, para onde são enviadas as amostras de sangue com o intuito de serem analisadas. Após a análise estar concluída, os resultados são devolvidos para a iGAM, que é responsável por os armazenar na base de dados referida anteriormente.

## Modelo de dados

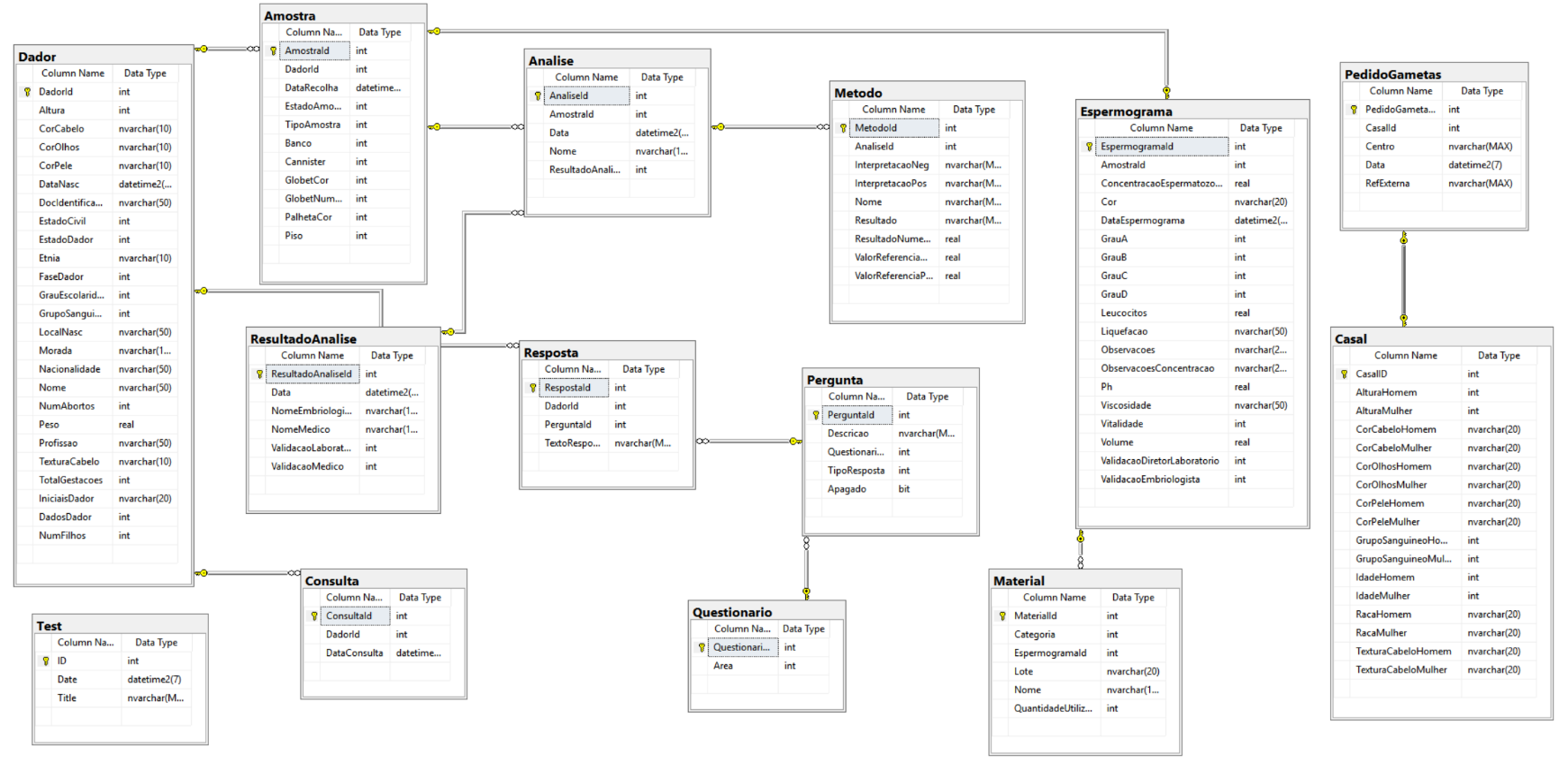


Figura 2 - Modelo de dados

Na Figura 2 é apresentado o modelo de dados do sistema que é constituído no total por 13 entidades. Como se pode observar, o **Dador** pode possuir várias **Amostras**, de sangue ou de espermatozoides, onde cada **Amostra** está associada apenas a um único **Dador**. O **Espermograma** por sua vez está associado a uma **Amostra** de espermatozoide, possuindo um conjunto de **Materiais** que foram usados. Para além disso, várias **Analises** de sangue dizem respeito a uma **Amostra**, sendo que as respetivas **Analises** fazem parte de um **ResultadoAnalise** e cada uma delas possui um conjunto de **Metodos**. Relativamente à entidade **Questionario**, cada **Questionario** médico possui um conjunto de **Perguntas** associadas, onde cada **Pergunta** possui múltiplas **Respostas**, pertencendo cada uma delas a um **Dador**. Este **Questionario** é respondido por um médico durante uma **Consulta** com o **Dador**, sendo que várias dessas **Consultas** estão associadas a um **Dador** único. Por fim, a última relação do modelo diz respeito ao **PedidoGametas** que está sempre relacionado com um único **Casal**.

## Diagrama de arquitetura

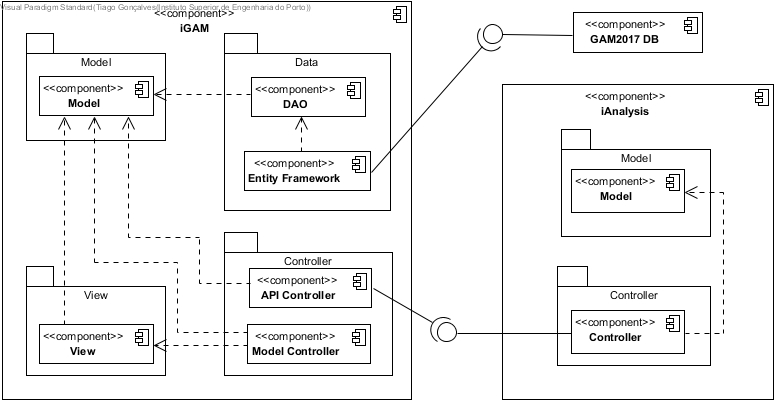


Figura 3 - Diagrama de arquitetura

A Figura 3 representa o Diagrama de Arquitetura da aplicação descrita ao longo deste relatório. É possível constatar que a solução apresentada é essencialmente composta por três componentes: a **iGAM**, a **iAnalysis** e a **GAM2017 DB**. A primeira, a **iGAM**, é o componente central de toda a solução. Assenta numa arquitetura MVC (Model View Controller), o que nos permite manter o código organizado, seguindo uma abordagem DRY (Don’t Repeat Yourself) e de “High Cohesion and Low Coupling”. O componente **GAM2017 DB**, representa a base de dados alojada na *cloud*, sendo a comunicação de dados mediada pela **Entity Framework** da Microsoft. Por fim, o componente **iAnalysis**, representa uma API que é responsável por analisar pedidos de análises de sangue, oriundo da plataforma iGAM. Este componente não tem persistência de dados e comunica sincronamente com a plataforma iGAM, isto é, recebe um pedido REST, processa os dados e devolve na resposta desse mesmo pedido os resultados da análise realizada.

# 5 Funcionalidades implementadas



## REQ 1 – Registo inicial do dador

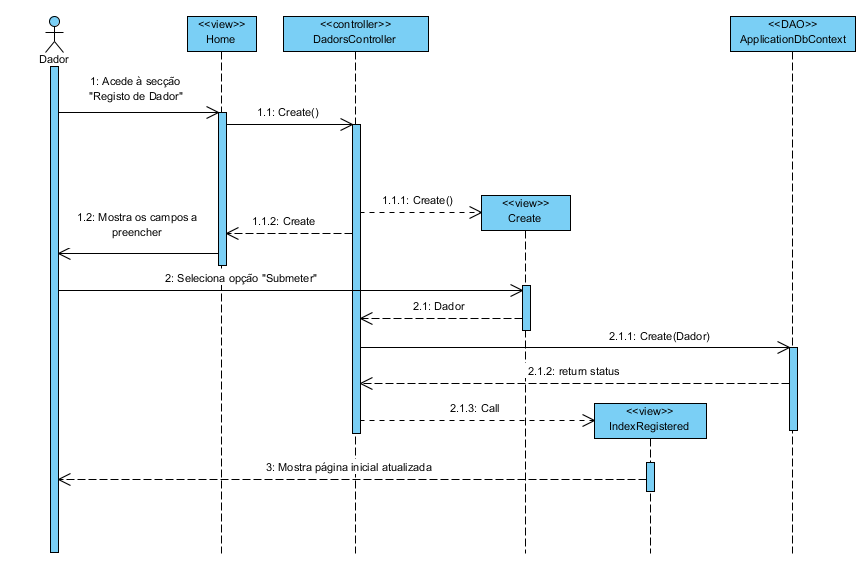
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 1 |
| Nome | Registo inicial do dador |
| Descrição | O Dador, enquanto utilizador não registado do sistema, pode submeter a sua candidatura através da página inicial da iGAM. Esta candidatura ficará pendente de aprovação. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Dador acede à secção de registo de um novo dador | 1.1 – A view “Home” executa o método “Create” do Controller “DadorsController”  1.2 – O Controller “DadorsController” carrega a view de “Dadors” associada, com os dados do dador por preencher |
| 2 – O Dador insere os dados solicitados | Sem ação do sistema |
| 3 – O Dador submete o registo | 3.1 – O Controller “DadorsController”, pelo método “Create”, insere o registo na base de dados  3.2 – O Controller “DadorsController” faz o *redirect* para a view “IndexRegistered” |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| Sem nada a registar | Os dados inseridos são validados pelo sistema | Os campos de texto não podem estar vazios |
|  | Os dados inseridos no sistema são armazenados numa base de dados | A idade deve ser um valor compreendido entre os 16 e os 60 anos |
|  | É gerado um identificador único para o(a) dador(a) | O NIF deve ser constituído por 9 dígitos |
|  |  | O contacto telefónico deve ser constituído por 9 dígitos |

### Diagrama de sequência



### Ilustração da funcionalidade



Figura 4 - REQ\_1: Página Inicial

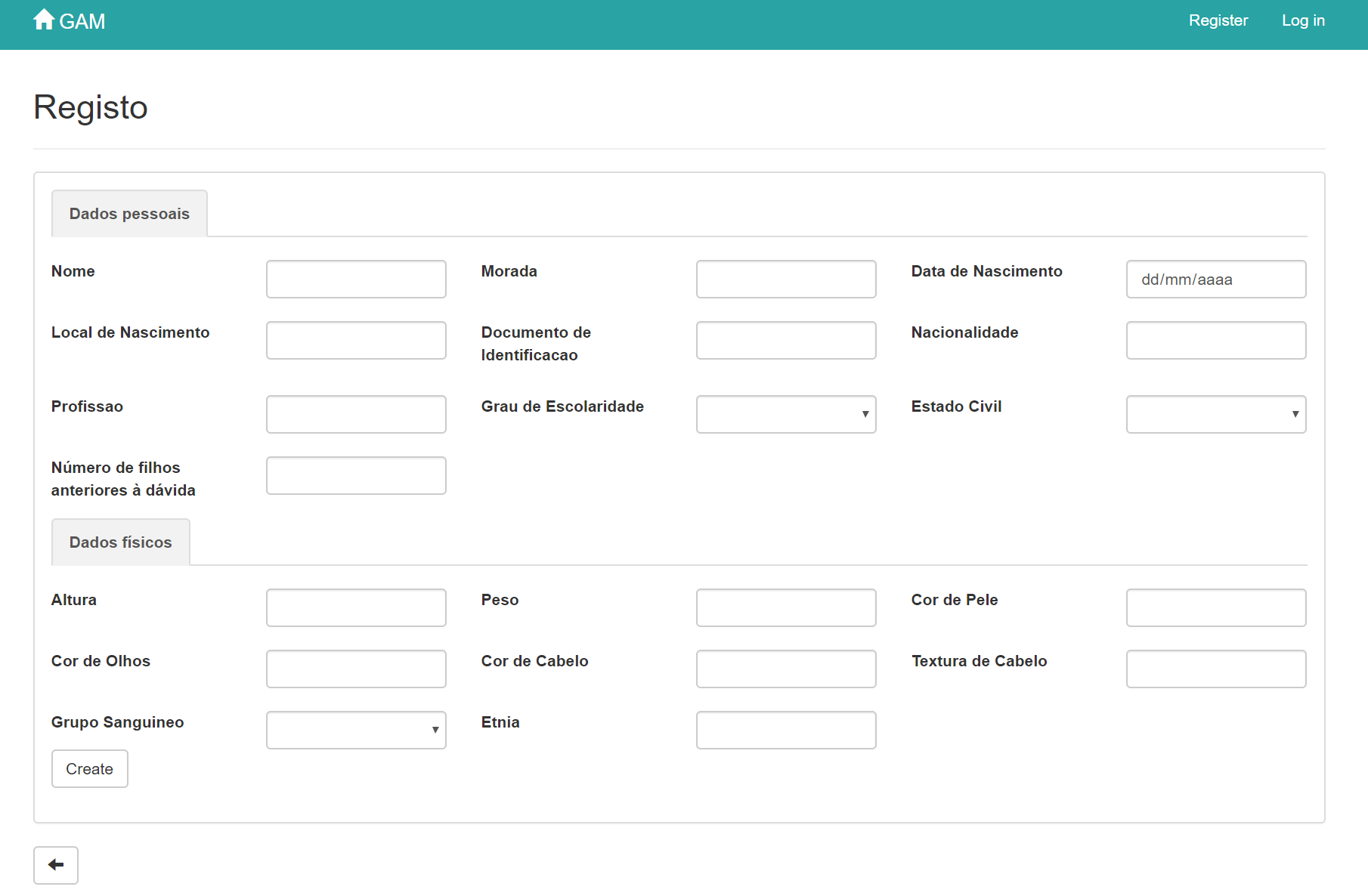


Figura 5 - REQ\_1: View de Registo do Dador

## REQ 2 – Validação dos dados do dador

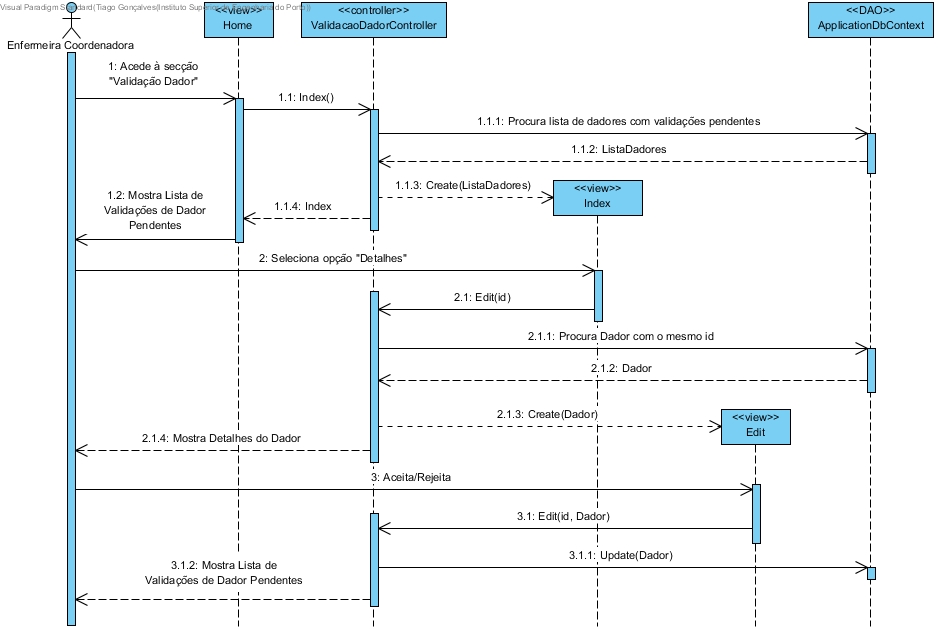
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 2 |
| Nome | Validação dos dados do dador |
| Descrição | A Enfermeira Coordenadora, enquanto utilizador registado do sistema, pode aceder à secção de “Validação Dador” na sua página da iGAM para validação de dados relativos a pedidos de registo de dadores. Esta validação poderá ser aceite ou rejeitada. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – A Enfermeira Coordenadora acede à secção “Validação Dador” | 1.1 – A view “Home” executa o método Index do Controller “ValidacaoDadorController”  1.2 – O Controller “ValidacaoDadorController” carrega a view de “ValidacaoDador” associada, com dadores que tenham dados pendentes para validação |
| 2 – A Enfermeira Coordenadora seleciona a opção “Detalhes” de um registo pendente | 2.1 – A view “ValidacaoDador” executa o método Edit do Controller “ValidacaoDadorController”  2.2 – O Controller “ValidacaoDadorController” carrega a view de “ValidacaoDador” associada, com os detalhes do dador selecionado |
| 3 – A Enfermeira Coordenadora aceita ou rejeita o registo | 3.1 – A view “ValidacaoDador” executa o método Edit do Controller “ValidacaoDadorController”  3.2 – O Controller “ValidacaoDadorController” atualiza o estado da validação dos dados do dador para aceite ou rejeitado conforme a escolha |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| A Enfermeira Coordenadora deve estar autenticada no sistema | O estado relativo à validação dos dados do dador é atualizado na base de dados | Sem nada a registar |

### Diagrama de sequência



### Ilustração da funcionalidade



Figura 6 - REQ\_2: View Lista de Dadores

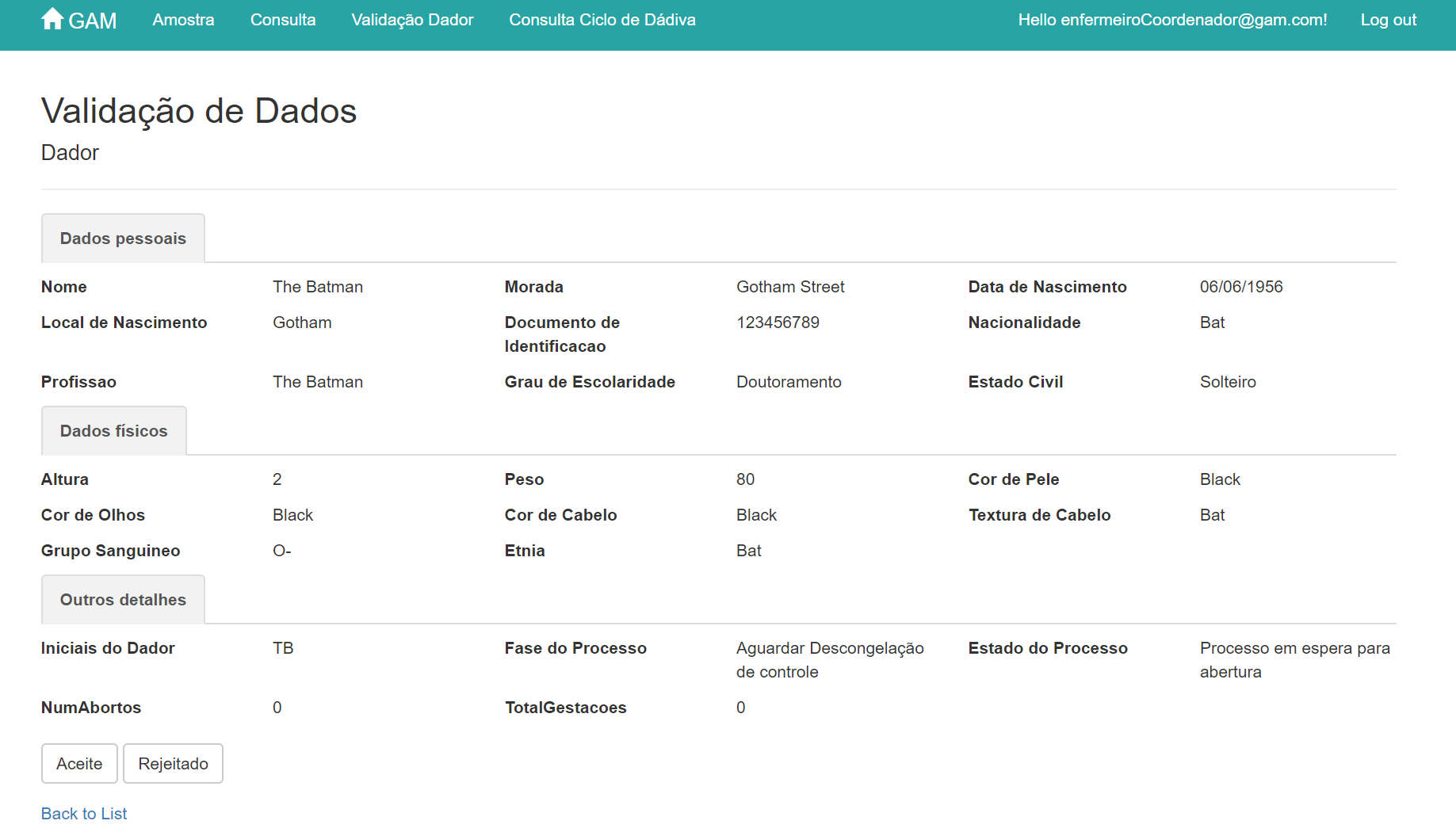


Figura 7 - REQ\_2: View da Validação de Dador

## REQ 3 – Gestão de perfis de utilizadores

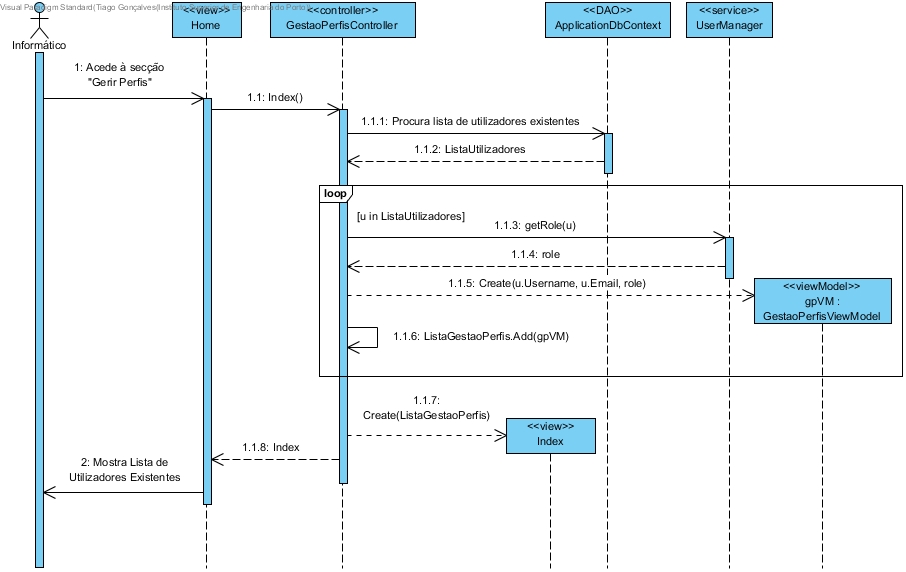
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 3 |
| Nome | Gestão de perfis de utilizadores |
| Descrição | O Informático, enquanto utilizador registado no sistema, pode aceder à secção “Gerir Perfis” na sua página da iGAM para criar, editar, ou eliminar utilizadores que podem aceder ao sistema, atribuindo um perfil respetivo ao cargo do utilizador. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Informático acede à secção “Gerir Perfis” | 1.1 – A view “Home” executa o método Index do Controller “GestaoPerfisController”  1.2 – O Controller “GestaoPerfisController” carrega a view de “GestaoPerfis” associada, com os utilizadores existentes no sistema |
| 2 – O Informático pode criar um novo utilizador, ou pode editar ou remover um utilizador existente | 2.1 – A view “GestaoPerfis” executa o método Create, Edit, ou Delete do Controller “GestaoPerfisController” consoante a respetiva opção  2.2 – O Controller “GestaoPerfisController” carrega a view de “GestaoPerfis” associada, com um formulário para criação ou edição do utilizador, ou um pedido de confirmação para eliminação da conta, consoante a respetiva opção |
| 3 – O Informático insere os dados para criação e edição, ou consulta a informação para remoção | Sem ação do sistema |
| 4 – O Informático confirma a ação | 4.1 – A view “GestaoPerfis” executa o método Create, Edit, ou DeleteConfirmed do Controller “GestaoPerfisController” consoante a respetiva ação  4.2 – O Controller “GestaoPerfisController” cria, atualiza, ou elimina os dados de um utilizador do sistema consoante a ação |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| O Informático deve estar autenticado no sistema | Os dados inseridos são validados pelo sistema (criação e edição) | Os campos de texto não podem estar vazios |
|  | Os dados inseridos no sistema são armazenados ou atualizados numa base de dados (criação e edição) | A password e a confirmação da password devem ser idênticas |
|  | Os dados são eliminados da base de dados (remoção) | A password deve conter um mínimo de 6 caracteres, um máximo 100 caracteres, letras maiúsculas, letras minúsculas, números e caracteres especiais |
|  | É gerado um identificador único para o utilizador (criação) |  |

### Diagrama de sequência



### Ilustração da funcionalidade

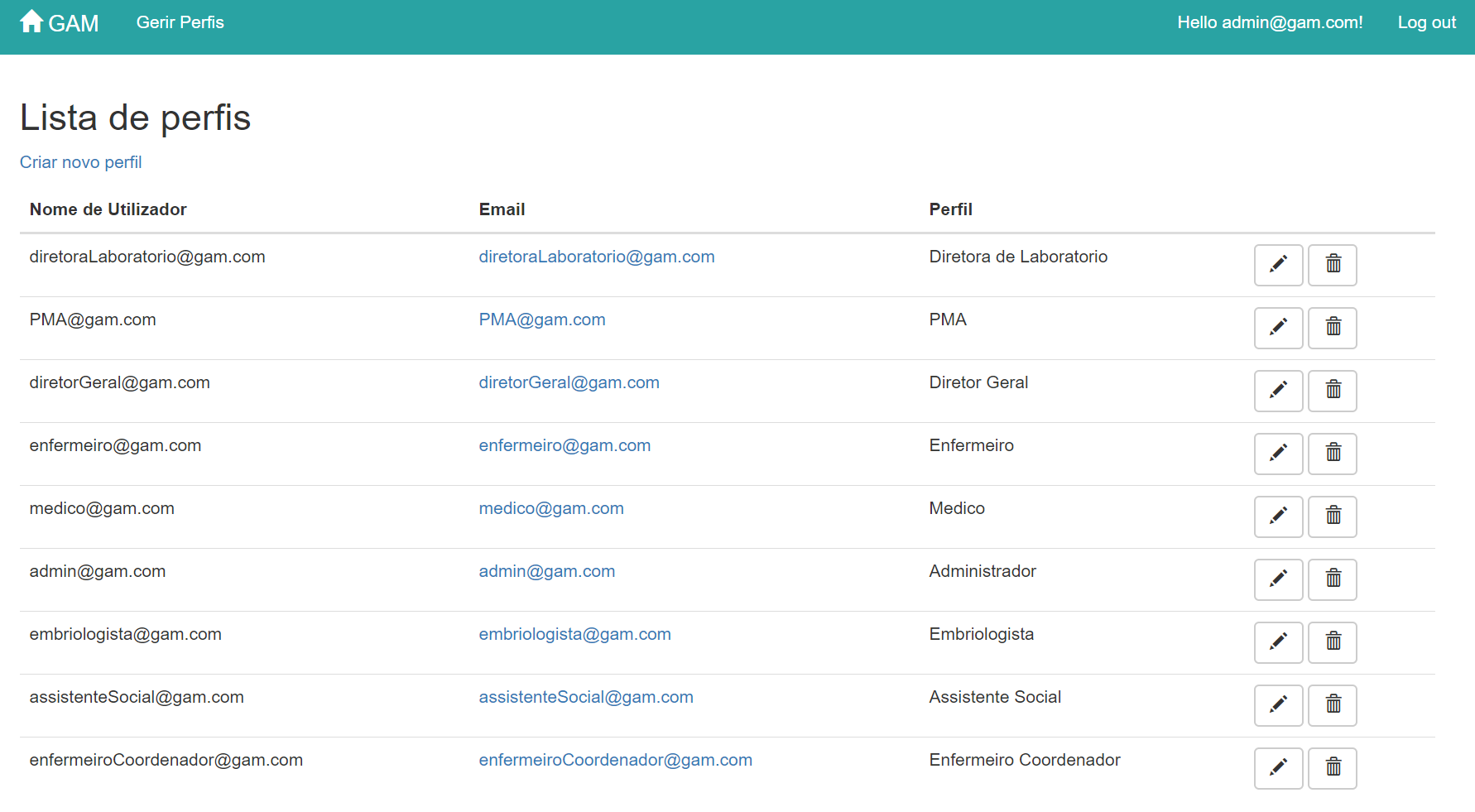


Figura 8 - REQ\_3: View da Lista de Perfis

Figura 9 - REQ\_3: View da Criação de um novo perfil

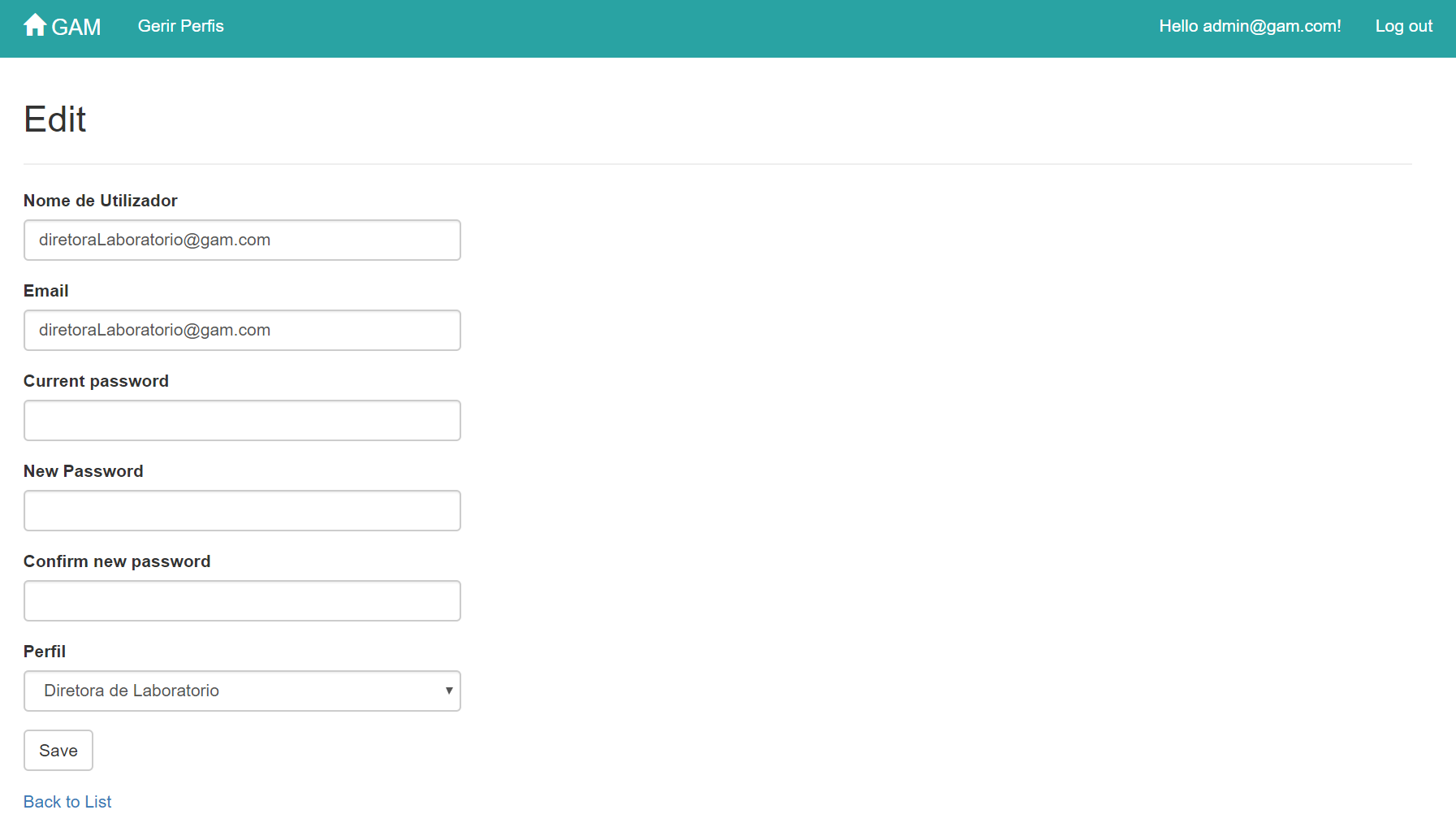


Figura 10: REQ\_3: View de edição de um perfil

## REQ 4 – Configuração do template de perguntas médicas

### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 4 |
| Nome | Configuração do template de perguntas médicas |
| Descrição | O Médico, enquanto utilizador registado no sistema, necessita de uma área onde possa personalizar um inquérito a realizar aos dadores. Nesse inquérito deve indicar a descrição das perguntas a efetuar, bem como o tipo de resposta que quer obter do dador. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Médico acede a opção do menu “configurar questionário” | * 1. – A view “Home” redireciona para o método “Index” do Controller “Questionario”   2. – O Controller, verifica a role do utilizador e carrega as perguntas configuradas para aquela role.   3. – O Controller devolve a view pronta para o utilizador poder manipular. |
| 2 – O Médico manipula o questionário e guarda as alterações | 2.1- A view emite um POST para o “QuestionarioController”, onde leva os dados do formulário  2.1 – Grava o questionário e respetivas perguntas para a BD  2.2 – Emite um *redirect* para o “Index”, do “HomeController” |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| Ser Médico ou Assistente social | Os dados inseridos são validados pelo sistema. | Existência de perguntas na BD |
|  | Os dados inseridos no sistema são armazenados numa base de dados. | Verifica se todas as perguntas têm tipos de perguntas |

### Diagrama de sequência

### Ilustração da funcionalidade

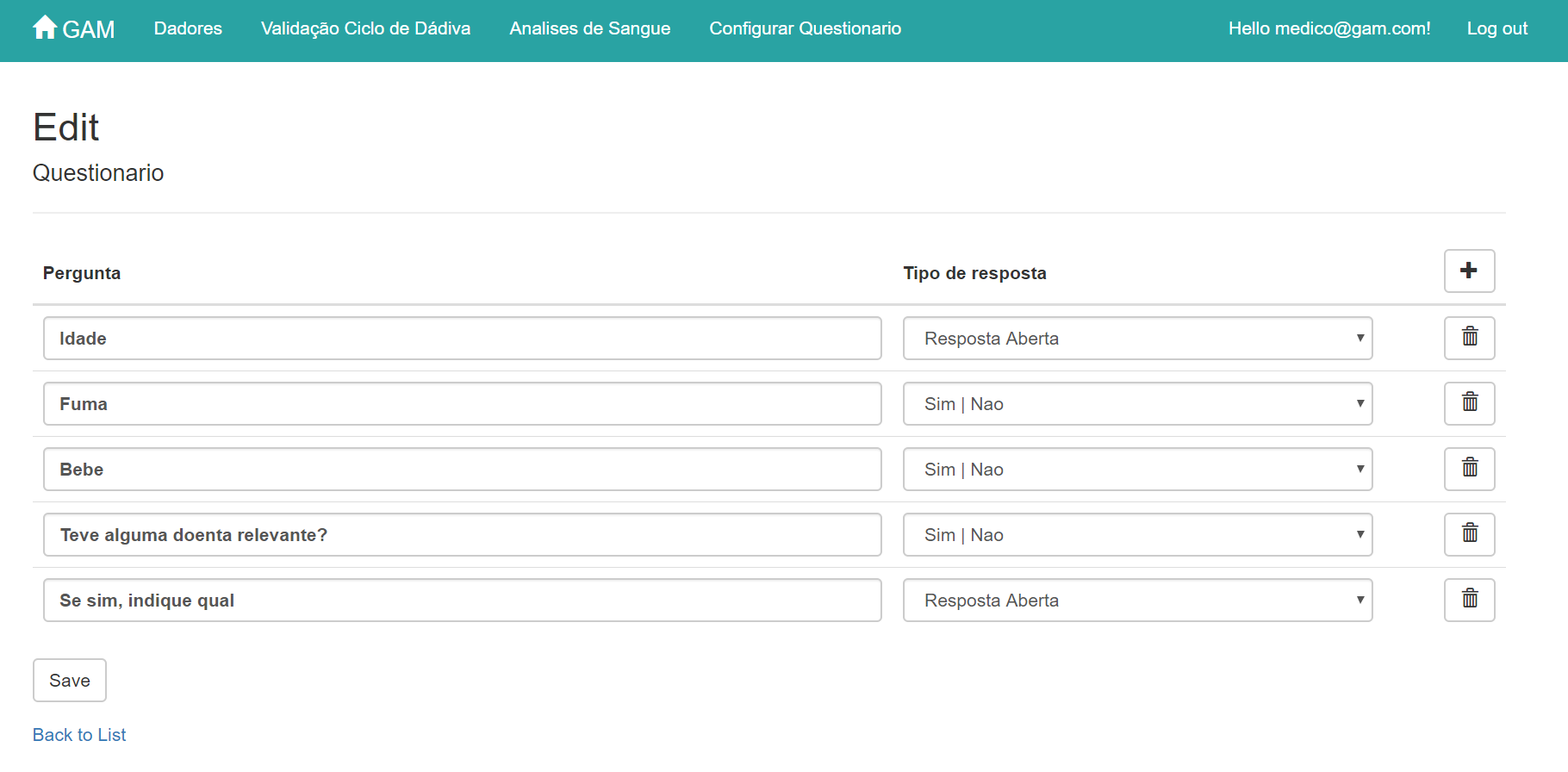


Figura 11 - REQ\_4: View de edição de um questionário

## REQ 5 – Consultas médicas via inquérito template

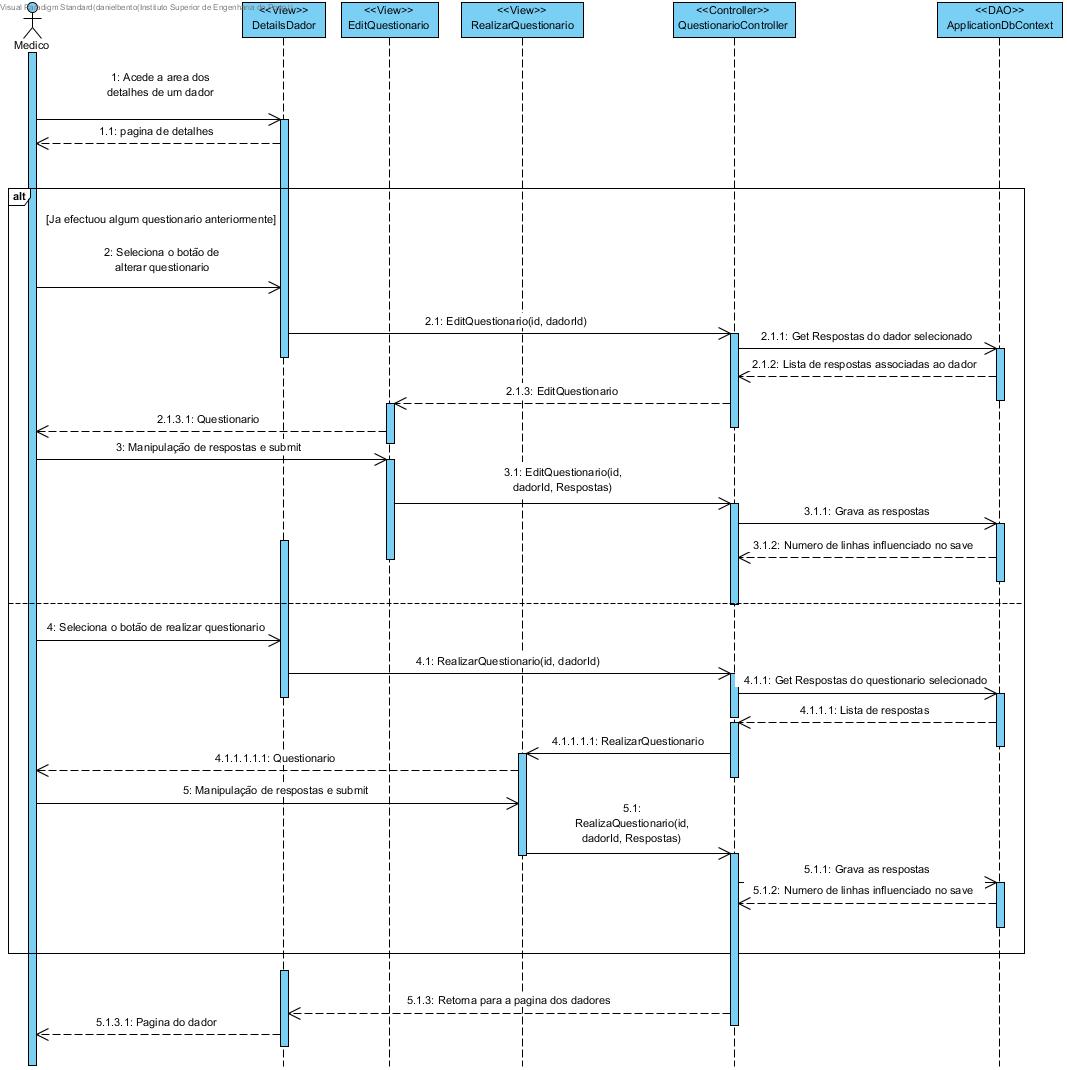
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 5 |
| Nome | Consultas médicas via inquérito template |
| Descrição | O Médico, enquanto utilizador registado do sistema, pode realizar um questionário a um dador, utilizando um template criado anteriormente |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Médico acede a área dos detalhes do dador | 1.1 – O Controller carrega os dados do dador e disponibiliza uma View com esses dados. |
| 2 – O Médico acede à secção do questionário, onde visualiza as perguntas realizadas anteriormente e onde poderá realizar um questionário. | 2.1- Sem ação do sistema |
| 3 – O Médico clica no botão “Realizar Questionário” ou “Alterar Questionário” | 3.1 - O “QuestionarioController” reúne as perguntas configuradas no template médico, e cria uma pagina com as perguntas |
| 4 – O Médico preenche o formulário | 4.1 – O sistema grava as perguntas, e retorna a View dos detalhes do dador selecionado. |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| Ser Médico | Os dados inseridos no sistema são armazenados numa base de dados. | Sem nada a registar |
| Ter um dador selecionado | Os dados inseridos são associados à ficha do dador. |  |

### Diagrama de sequência



### Ilustração da funcionalidade



Figura 12 - REQ\_5: View da lista de possíveis dadores

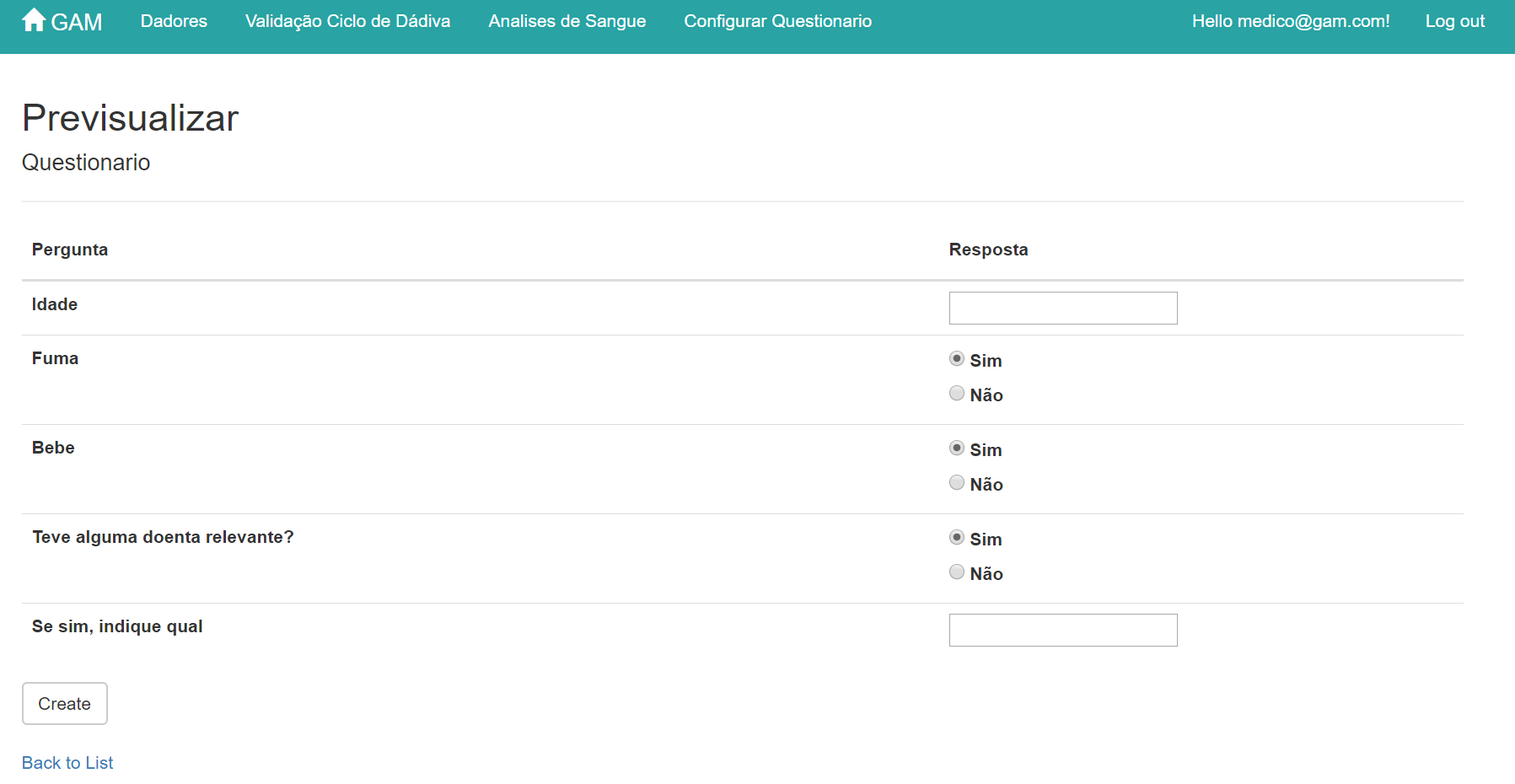


Figura 13: REQ\_5: View do preenchimento do formulário



Figura 14 - REQ5: View dos detalhes do dador

## REQ 6 – Registo de amostras

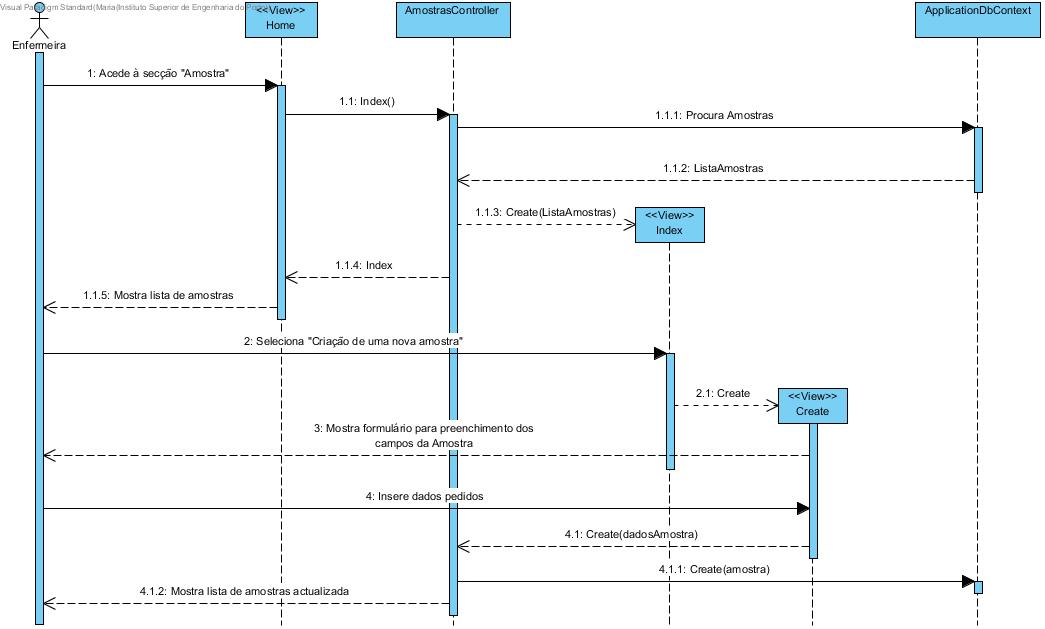
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 6 |
| Nome | Registo de amostras |
| Descrição | A Enfermeira, enquanto utilizadora registada no sistema, pode aceder à secção “Amostras” na sua página da iGAM para registar espermogramas. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – A Enfermeira acede à secção “Amostra”. | 1.1 – A view “Home” executa o método Index do Controller “AmostrasController”  1.2 – O Controller “AmostrasController” carrega a view de “Amostras” associada, com as amostras existentes no sistema. |
| 2 – A Enfermeira acede à “criação de uma nova amostra”. | 2.1 – A view “Amostras” executa o método Create do Controller “AmostrassController”  2.2 O Controller “AmostrasController” carrega a view de “Amostras” associada, com um formulário para criação. |
| 3 – A Enfermeira visualiza a lista de amostras criadas atualizada. | 3.1 –O Controller “AmostrasController” carrega a view de “Amostras” associada, com as amostras existentes no sistema. |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| O Embriologista deve estar autenticada no sistema | A amostra é registada na base de dados | Sem nada a registar |

### Diagrama de sequência



### Ilustração da funcionalidade



Figura 15 - REQ\_6: View da lista de amostras

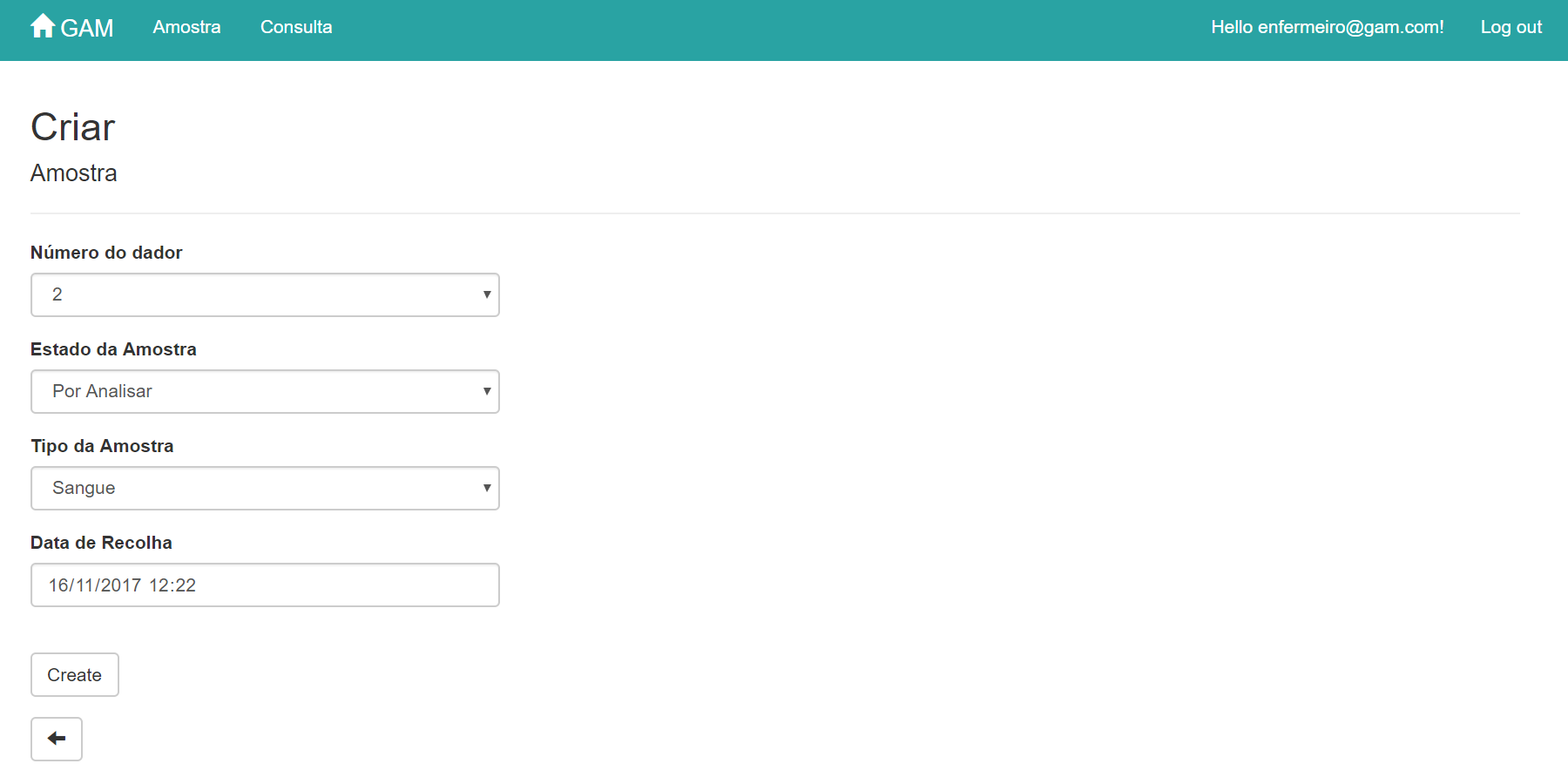


Figura 16 - REQ\_6: View de criação de uma amostra



Figura 17 - REQ\_6: View de edição de uma amostra

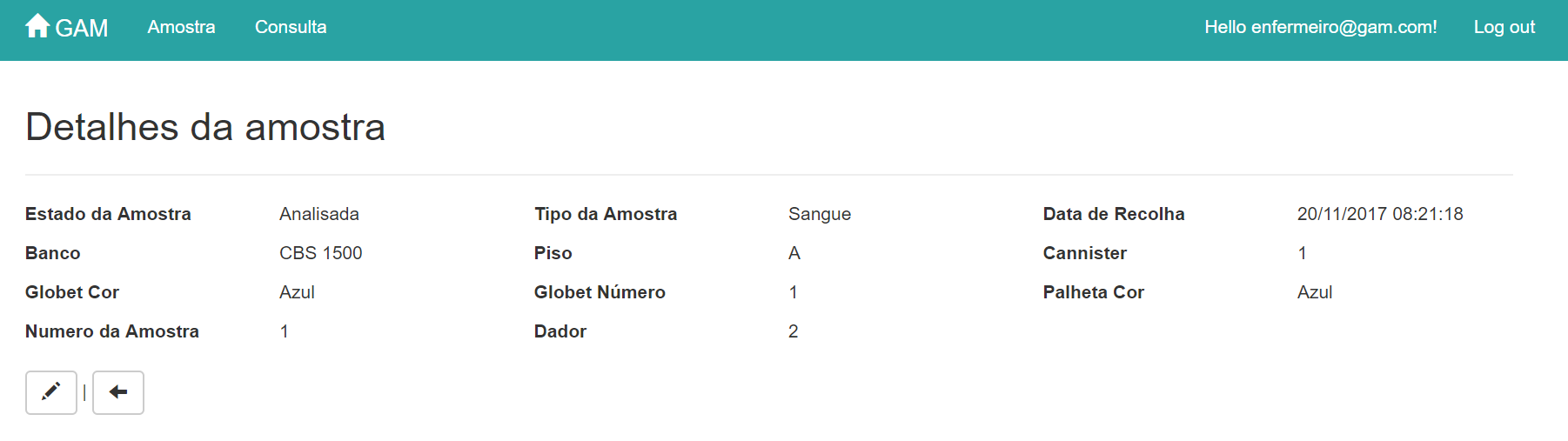


Figura 18 - REQ\_6 - View de detalhes de uma amostra

## REQ 7 – Pedido das análises de sangue iniciais

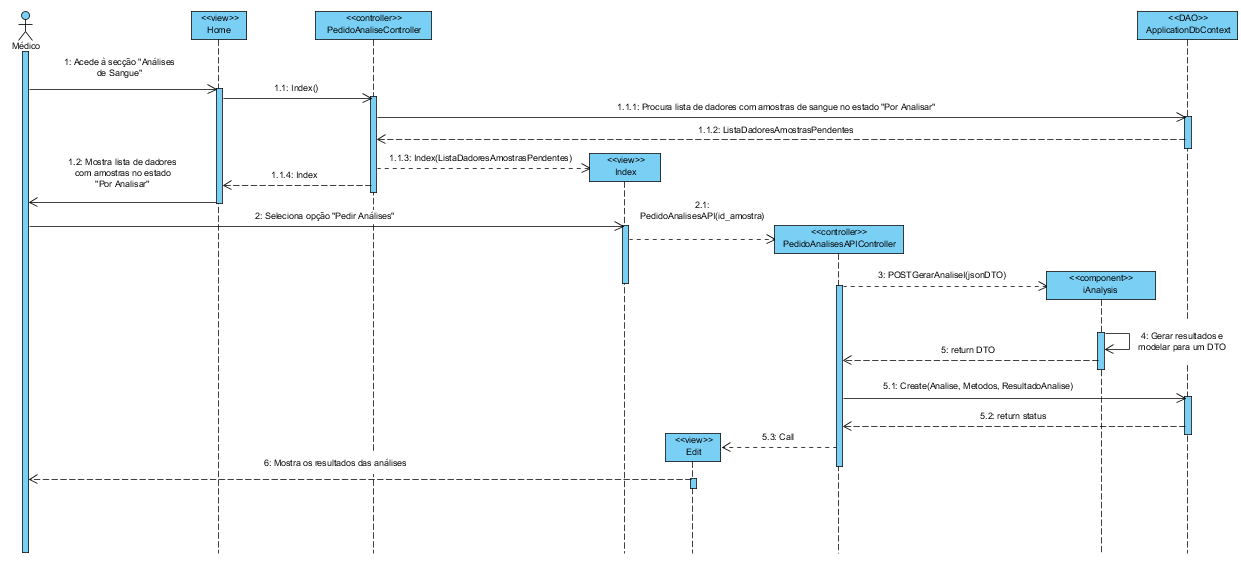
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 7 |
| Nome | Pedido das análises de sangue iniciais |
| Descrição | O Médico, enquanto utilizador registado do sistema, pode aceder à secção “Análises de Sangue” na sua página iGAM para submeter um pedido de análises de sangue ao Laboratório de Análises. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Médico acede à secção “Análises de Sangue” | 1.1 – A view “Home” executa o método Index do Controller “PedidoAnaliseController”  1.2 – O Controller “PedidoAnaliseController” carrega a view “PedidoAnalise”, com dadores que têm amostras de sangue por analisar. |
| 2 – O Médico seleciona a opção “Pedir Análises” | 2.1 – A view “PedidoAnalise” executa o método “PedidoAnalisesAPI” do Controller “PedidoAnalisesAPIController”  2.2 – O Controller “PedidoAnalisesAPIController” estabelece a conexão por um pedido REST com a API externa iAnalysis.  2.3 – A API iAnalysis recebe o identificador da amostra a ser analisada e gerar os resultados para todos os parâmetros que compõe uma análise. Esses resultados são devolvidos na forma de um DTO para o Controller “PedidoAnalisesAPIController”.  2.4 – O Controller “PedidoAnalisesAPIController” modela os dados retornados pela API e guarda-os na base de dados. |
| 3 – O Médico consulta os resultados das análises | 3.1 – O Controller “PedidoAnalisesAPIController” invoca a view “Edit” que mostra os resultados das análises |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| Devem existir amostras do tipo Sangue no estado “Por Analisar” | Os dados devolvidos pela API iAnalysis devem ser armazenados numa base de dados | Sem nada a registar |
|  | Os dados devolvidos pela API iAnalysis devem ser associados à ficha do dador |  |

### Diagrama de sequência



### Ilustração da funcionalidade

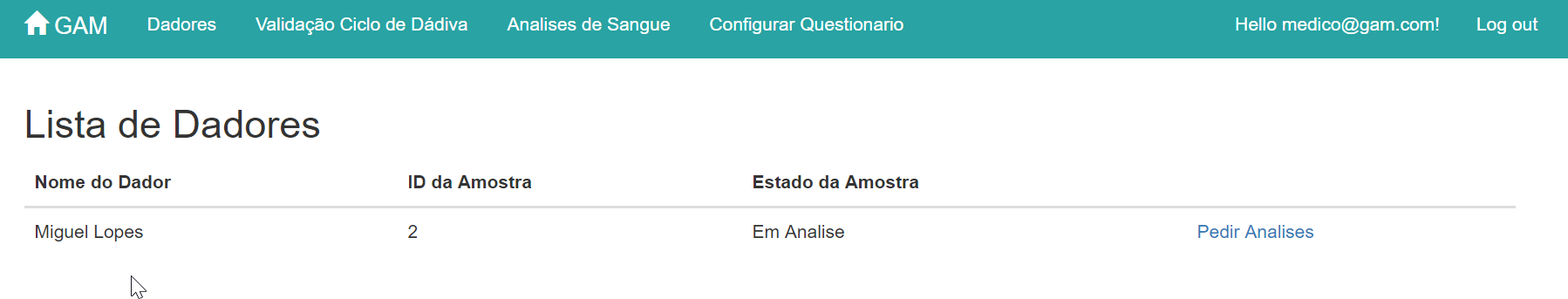


Figura 19 - REQ\_7: View da listagem de dadores com amostras pendentes de análise

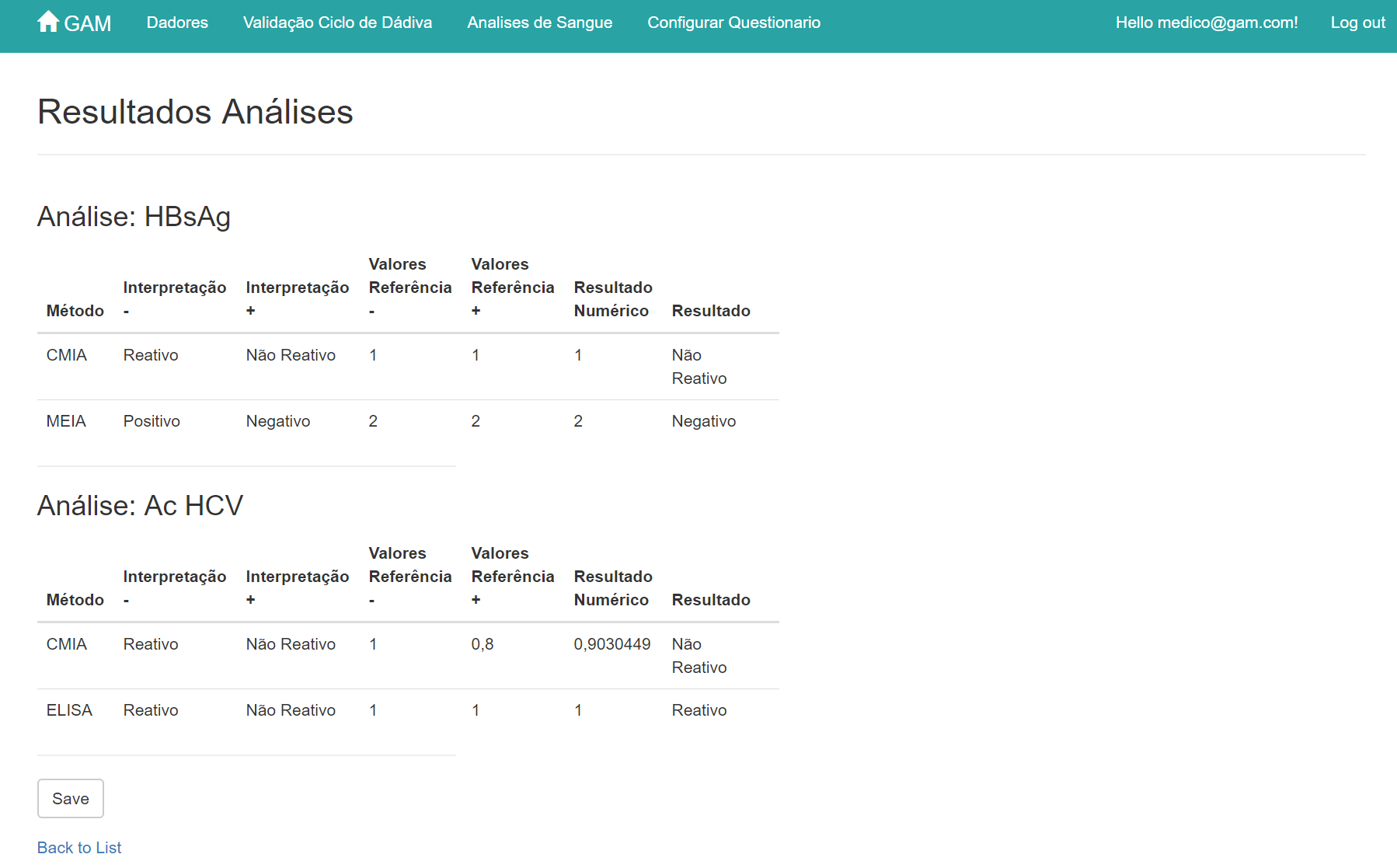


Figura 20 - REQ\_7: View dos resultados das análises gerados pela API iAnalysis

## REQ 8 – Ciclo da dádiva (abertura e consulta de informação)

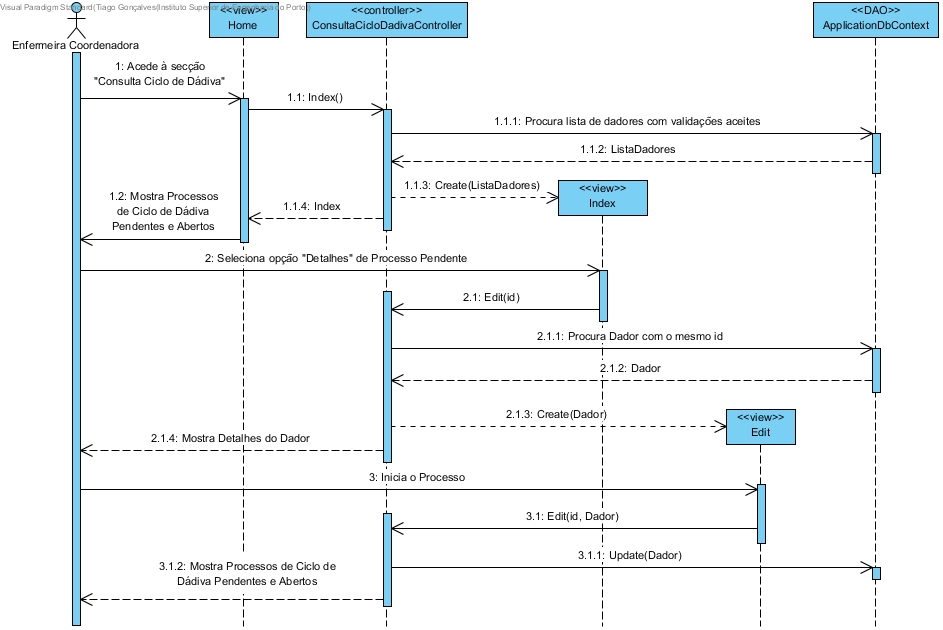
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 8 |
| Nome | Ciclo da dádiva (abertura e consulta de informação) |
| Descrição | A Enfermeira Coordenadora enquanto utilizadora registada no sistema, pode aceder à secção “Consulta Ciclo de Dádiva” na sua página da iGAM para visualizar processos pendentes que podem ser iniciados, e processos abertos |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – A Enfermeira Coordenadora acede à secção “Consulta Ciclo de Dádiva” | 1.1 – A view “Home” executa o método Index do Controller “ConsultaCicloDadivaController”  1.2 – O Controller “ConsultaCicloDadivaController” carrega a view de “ConsultaCicloDadiva” associada, com os ciclos de dadiva pendentes e aceites |
| 2 – A Enfermeira Coordenadora seleciona um ciclo de dádiva pendente ou aberto | 2.1 – A view “ConsultaCicloDadiva” executa o método Edit ou Details do Controller “ConsultaCicloDadivaController” consoante o caso  2.2 – O Controller “ConsultaCicloDadivaController” carrega a view de “ConsultaCicloDadiva” associada, com informação sobre o dador do respetivo ciclo de dádiva |
| 3 – A Enfermeira Coordenadora inicia o processo de ciclo de dádiva para ciclos pendentes | 3.1 – A view “ConsultaCicloDadiva” executa o método Edit do Controller “ConsultaCicloDadivaController”  3.2 – O Controller “ConsultaCicloDadivaController” atualiza o estado do processo do dador |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| A Enfermeira Coordenadora deve estar autenticada no sistema | O estado do processo do dador é atualizado na base de dados | Sem nada a registar |

### Diagrama de sequência



### Ilustração da funcionalidade

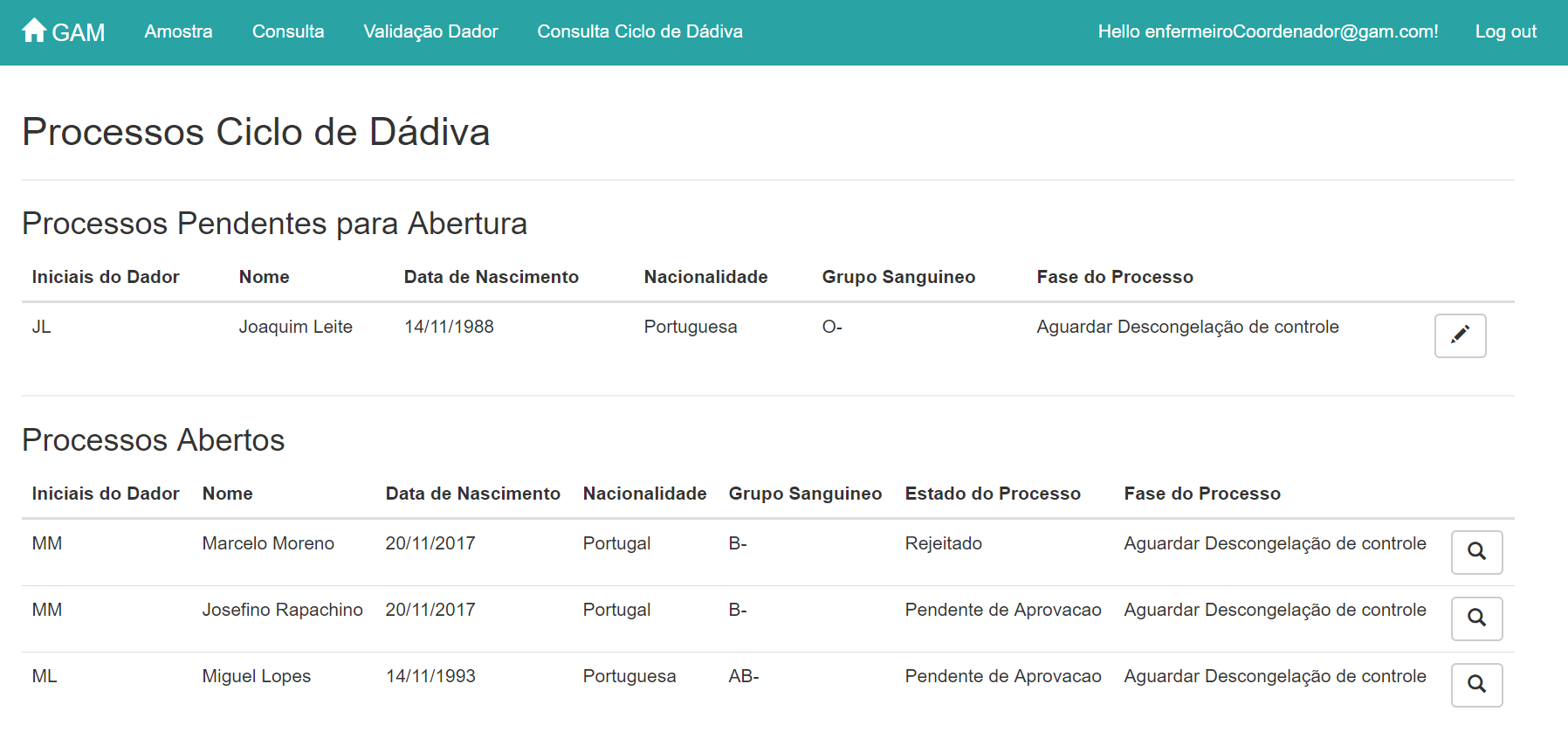


Figura 21 - REQ\_8: View dos processos pendentes e abertos

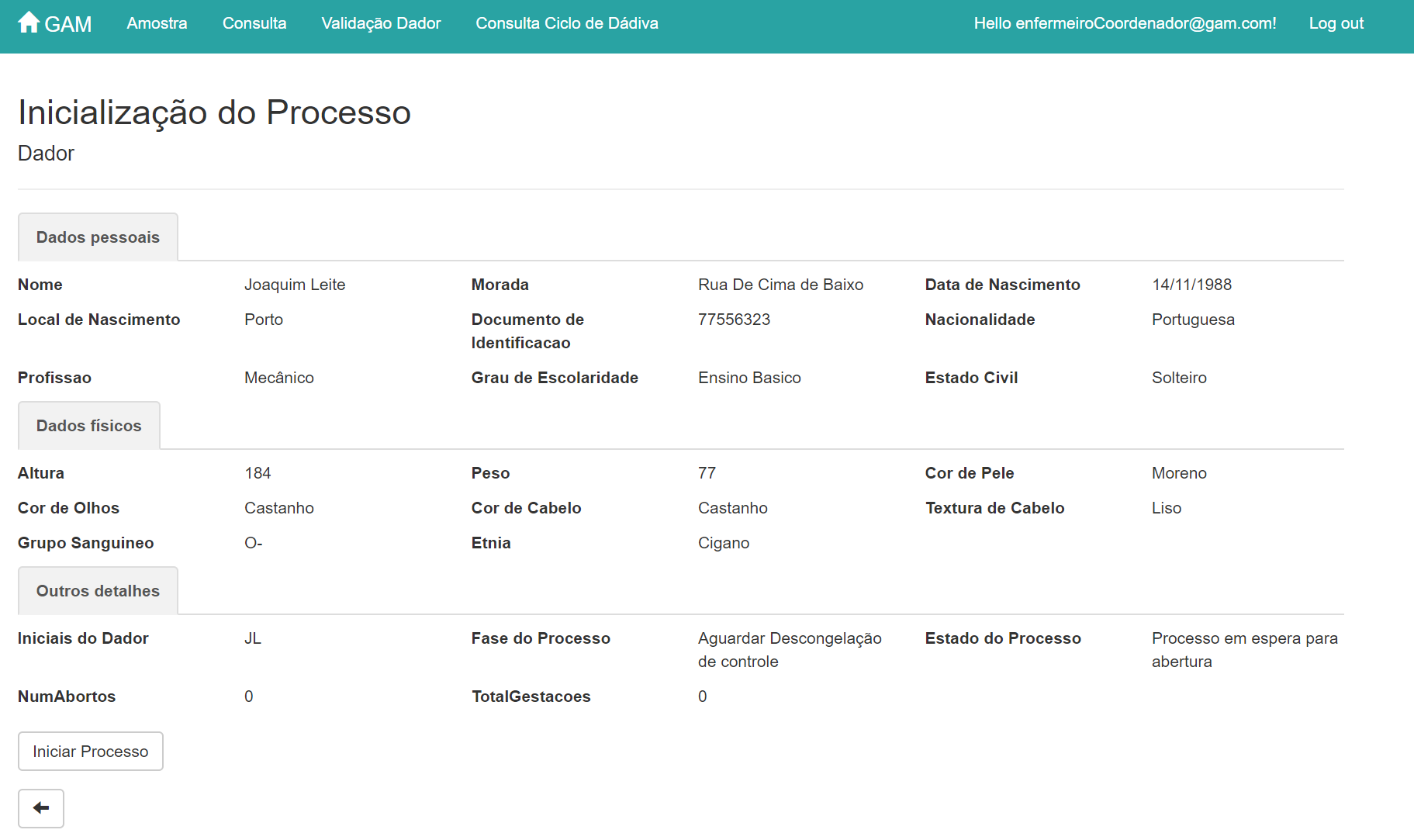


Figura 22 - REQ\_8: View de abertura do processo

## REQ 9 – Marcação de consultas

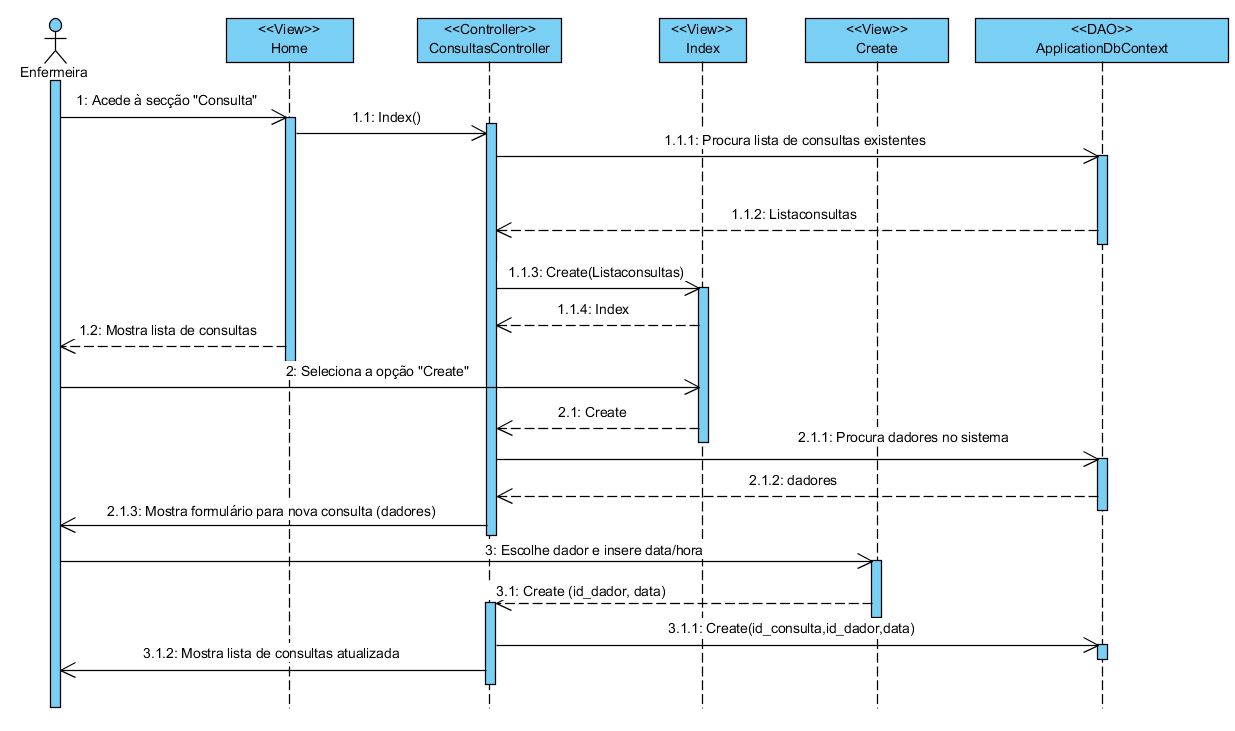
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 9 |
| Nome | Marcação de consultas |
| Descrição | A enfermeira, enquanto utilizador registado no sistema, pode aceder à secção “Marcação de Consultas” na sua página da iGAM para criar, editar, ou eliminar consultas ao dador. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – A Enfermeira acede à secção “Consulta” | 1.1 – A view “Home” executa o método Index do Controller “ConsultasController”  1.1.2 – O Controller “ConsultasController” carrega a view de “Consultas” associada, com as consultas existentes no sistema |
| 2 – A Enfermeira escolhe uma das opções: criar nova consulta\*, editar consulta ou remover consulta. | 2.1 – A view “Consultas” executa os métodos Create, Edit, ou Delete do Controller “ConsultasController” consoante a respetiva opção  2.1.1 – O Controller “ConsultasController” carrega a lista de dadores do sistema  2.1.3 – O Controller “ConsultasController” carrega a view de “Consultas” associada, com um formulário para criação ou edição da consulta, ou um pedido de confirmação para eliminação da consulta, consoante a respetiva opção |
| 3 – A Enfermeira visualiza o formulário de criação de nova consulta. Escolhe o dador e insere a data e hora | 3.1 – A view “Create” executa o método Create do Controller “ConsultasController”  3.1.1 - O Controller “ConsultasController” executa o método Create(id\_consulta,id\_dador,data) da ApplicationDbContext.  3.1.2 – O sistema apresenta a lista de consultas atualizada |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| A Enfermeira Coordenadora deve estar autenticada no sistema | A consulta é registada/atualizada/removida da base de dados | A data da consulta deve ser posterior à data atual |
| O dador deve estar criado no sistema |  |  |

### Diagrama de sequência



### Ilustração da funcionalidade

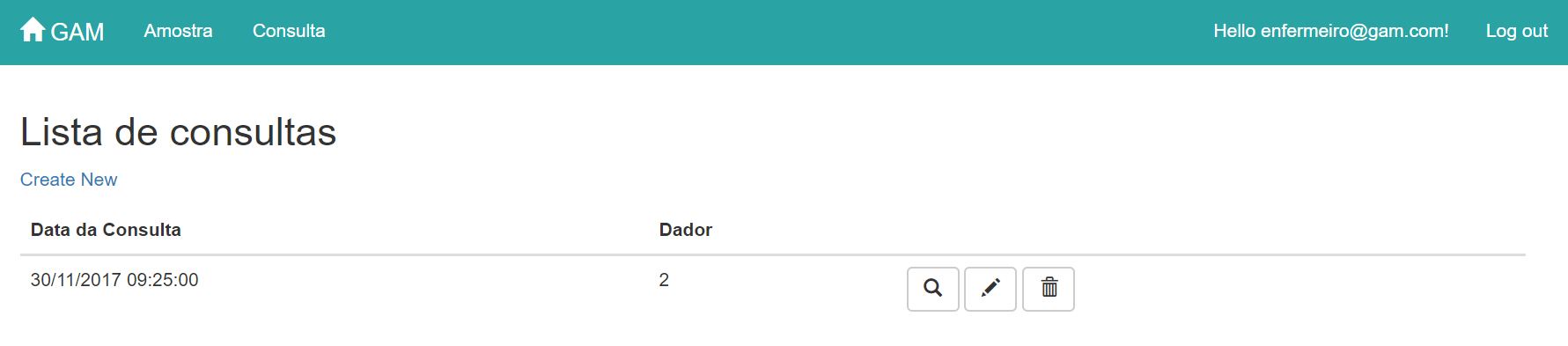


Figura 23 - REQ\_9: View da lista de consultas

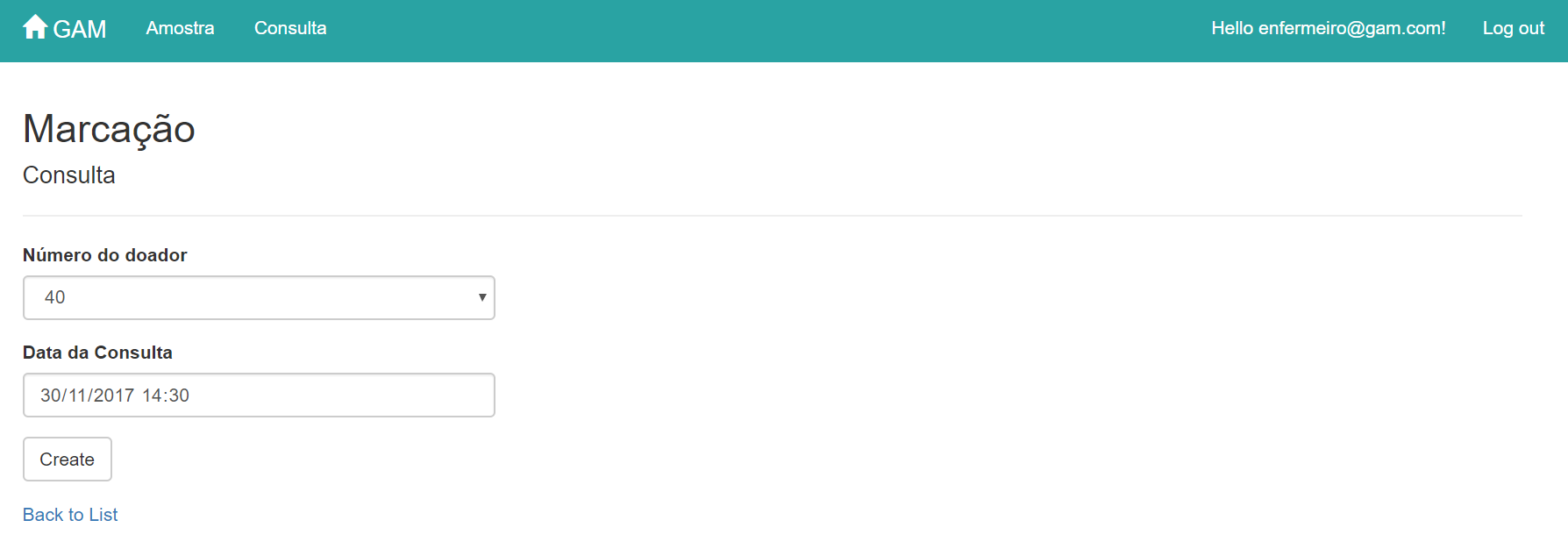


Figura 24 - REQ\_9: View da marcação de uma consulta

## REQ 10 – Aprovação/Rejeição do ciclo da dádiva

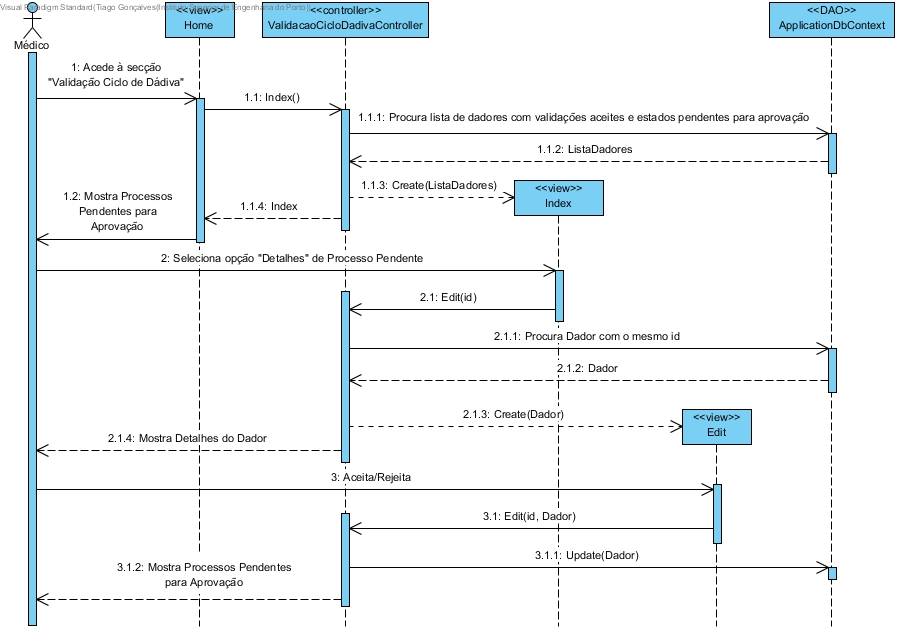
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 10 |
| Nome | Aprovação/Rejeição do ciclo da dádiva |
| Descrição | O Médico enquanto utilizador registado no sistema, pode aceder à secção “Validação Ciclo de Dádiva” na sua página da iGAM para visualizar processos de ciclo de dádiva pendentes para aprovação. Estes processos podem ser aceites ou rejeitados. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Médico acede à secção “Validação Ciclo de Dádiva” | 1.1 – A view “Home” executa o método Index do Controller “ValidacaoCicloDadivaController”  1.2 – O Controller “ValidacaoCicloDadivaController” carrega a view de “ValidacaoCicloDadiva” associada, com os ciclos de dadiva pendentes para aprovação |
| 2 – O Médico seleciona um processo de ciclo de dádiva pendente para aprovação | 2.1 – A view “ValidacaoCicloDadiva” executa o método Edit do Controller “ValidacaoCicloDadivaController”  2.2 – O Controller “ValidacaoCicloDadivaController” carrega a view de “ValidacaoCicloDadiva” associada, com informação sobre o dador do respetivo ciclo de dádiva |
| 3 – O Médico aceita ou rejeita o processo de ciclo de dádiva selecionado | 3.1 – A view “ValidacaoCicloDadiva” executa o método Edit do Controller “ValidacaoCicloDadivaController”  3.2 – O Controller “ValidacaoCicloDadivaController” atualiza o estado do processo do dador para aceite ou rejeitado consoante a opção selecionada |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| O Médico deve estar autenticado no sistema | O estado do processo do dador é atualizado na base de dados | Sem nada a registar |

### Diagrama de sequência



### Ilustração da funcionalidade

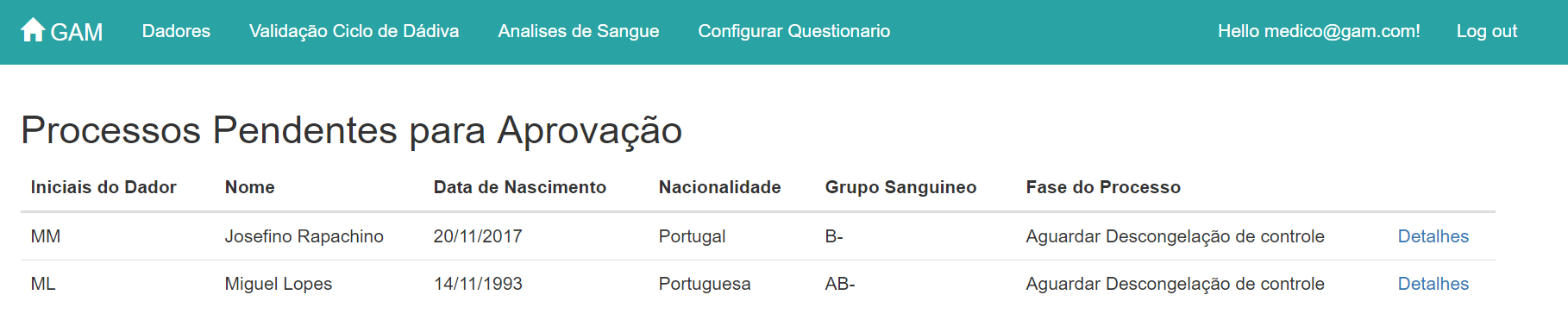


Figura 25 - REQ\_10: View da lista de processos pendentes de aprovação



Figura 26 - REQ\_10: View da aprovação/rejeição de um processo

## REQ 11 – Lista de trabalhos do laboratório

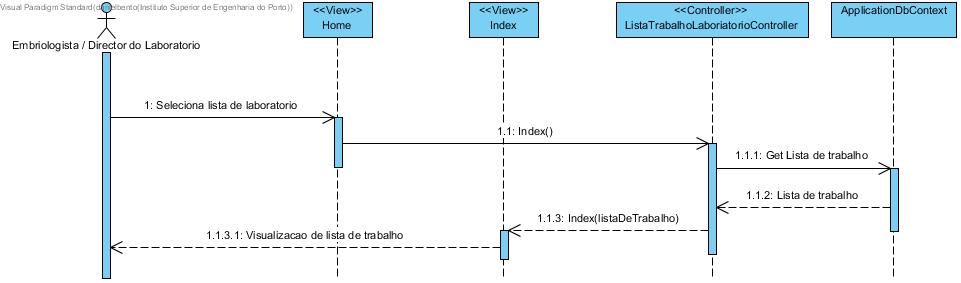
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 11 |
| Nome | Lista de trabalhos do laboratório |
| Descrição | O Embriologista/Diretora de Laboratório, enquanto utilizadores registados do sistema, podem visualizar a lista de trabalhos pendentes no laboratório |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Embriologista ou Diretora de Laboratório acede a área da lista de trabalhos do laboratório | 1.1 – O “Index” do Controller “ListaDeLaboratorio” seleciona os trabalhos pendentes, e disponibiliza uma View com os estados agrupados por tipo de estado do trabalho. |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| Ter sessão iniciada como Embriologista ou Diretora de Laboratório | Sem nada a registar | Sem nada a registar |

### Diagrama de sequência



### Ilustração da funcionalidade



Figura 27 - REQ\_11: View da lista de amostras por analisar



Figura 28 - REQ\_11: View da lista de amostras analisadas

## REQ 12 – Registo de espermograma

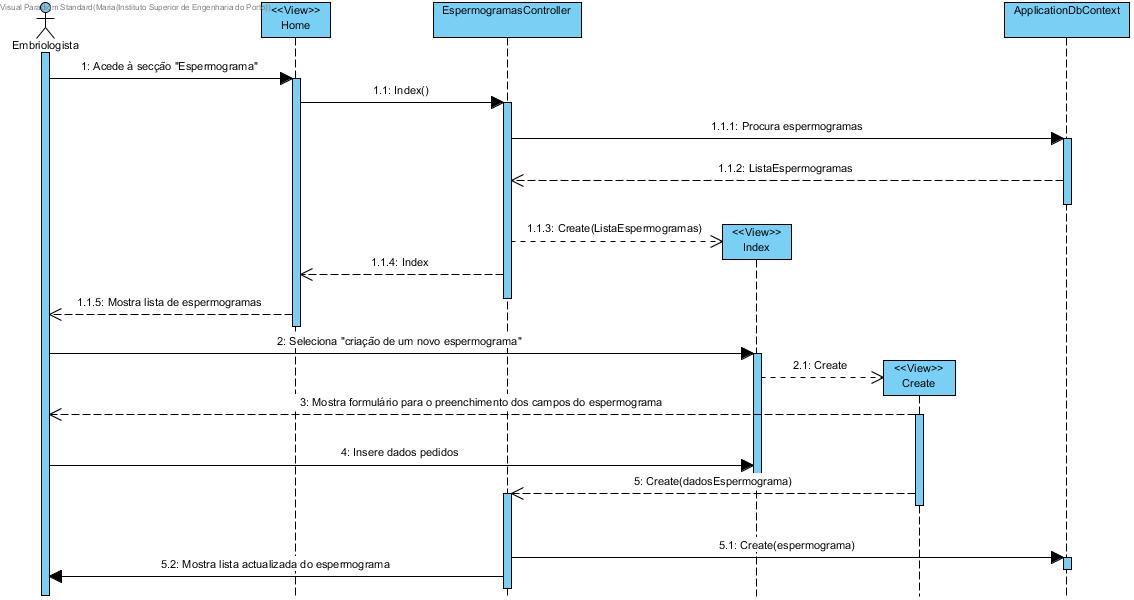
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 12 |
| Nome | Registo de espermograma |
| Descrição | O Embriologista, enquanto utilizador registado no sistema, pode aceder à secção “Material” na sua página da iGAM para criar um espermograma. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Embriologista acede à secção “Espermograma” | 1.1 – A view “Home” executa o método Index do Controller “EspermogramasController”  1.2 – – O Controller “EspermogramasController” carrega a view de “Espermogramas” associada, com os espermogramas existentes no sistema. |
| 2 – O Embriologista acede à “criação de um novo espermograma. | 2.1 –A view “Espermogramas” executa o método Create do Controller “EspermogramasController”  2.2 – O Controller “EspermogramasController” carrega a view de “Espermograma” associada, com um formulário para criação. |
| 3 – O Embriologista visualiza a lista de espermogramas atualizada. | 3.1 – O Controller “EspermogramasController” carrega a view de “Espermograma” associada, com os espermogramas existentes no sistema. |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| O Embriologista deve estar autenticada no sistema | O registo do espermograma é atualizado na base de dados | Sem nada a registar |

### Diagrama de sequência



### Ilustração da funcionalidade

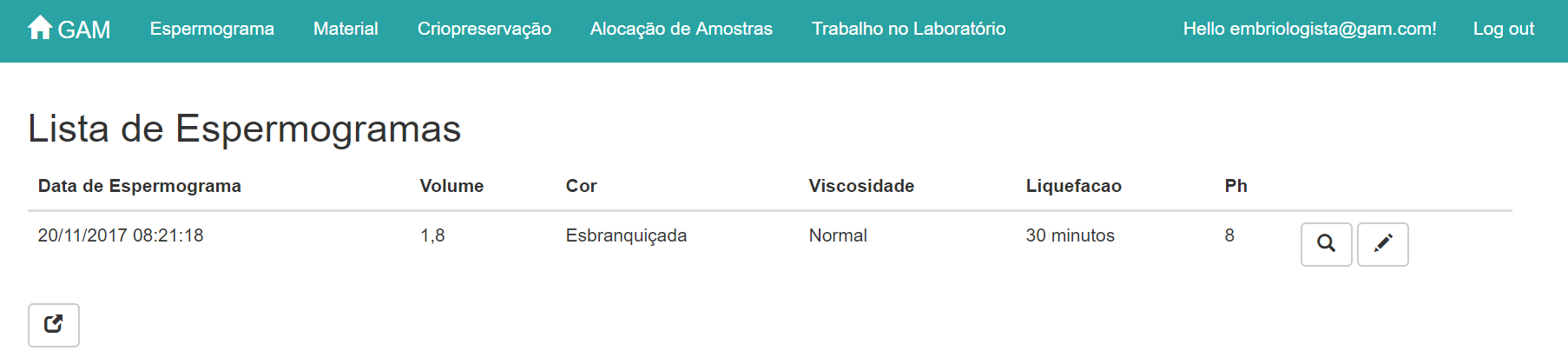


Figura 29 - REQ\_12 – View da lista de espermogramas

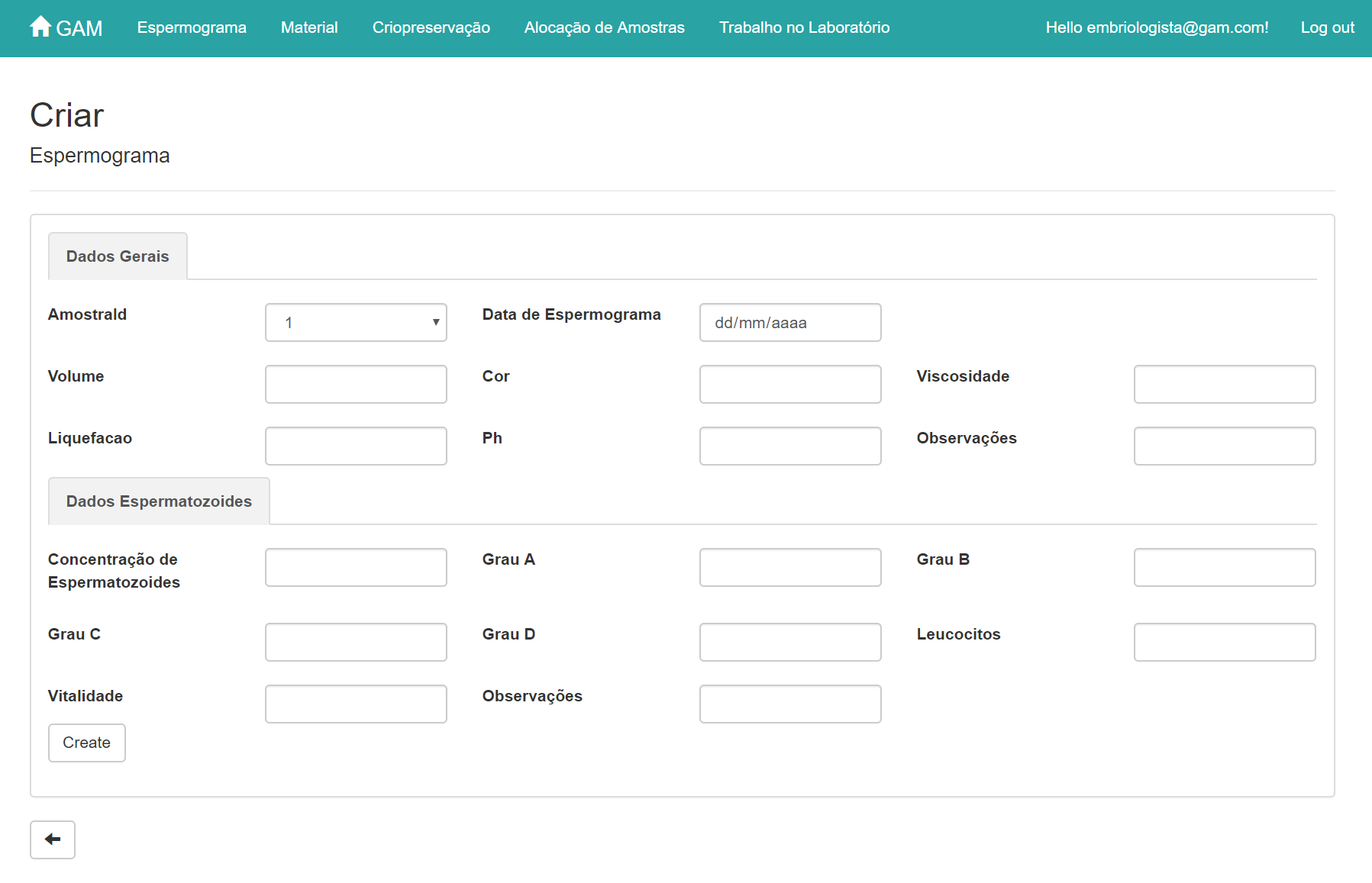


Figura 30 - REQ\_12: View de criação de espermograma

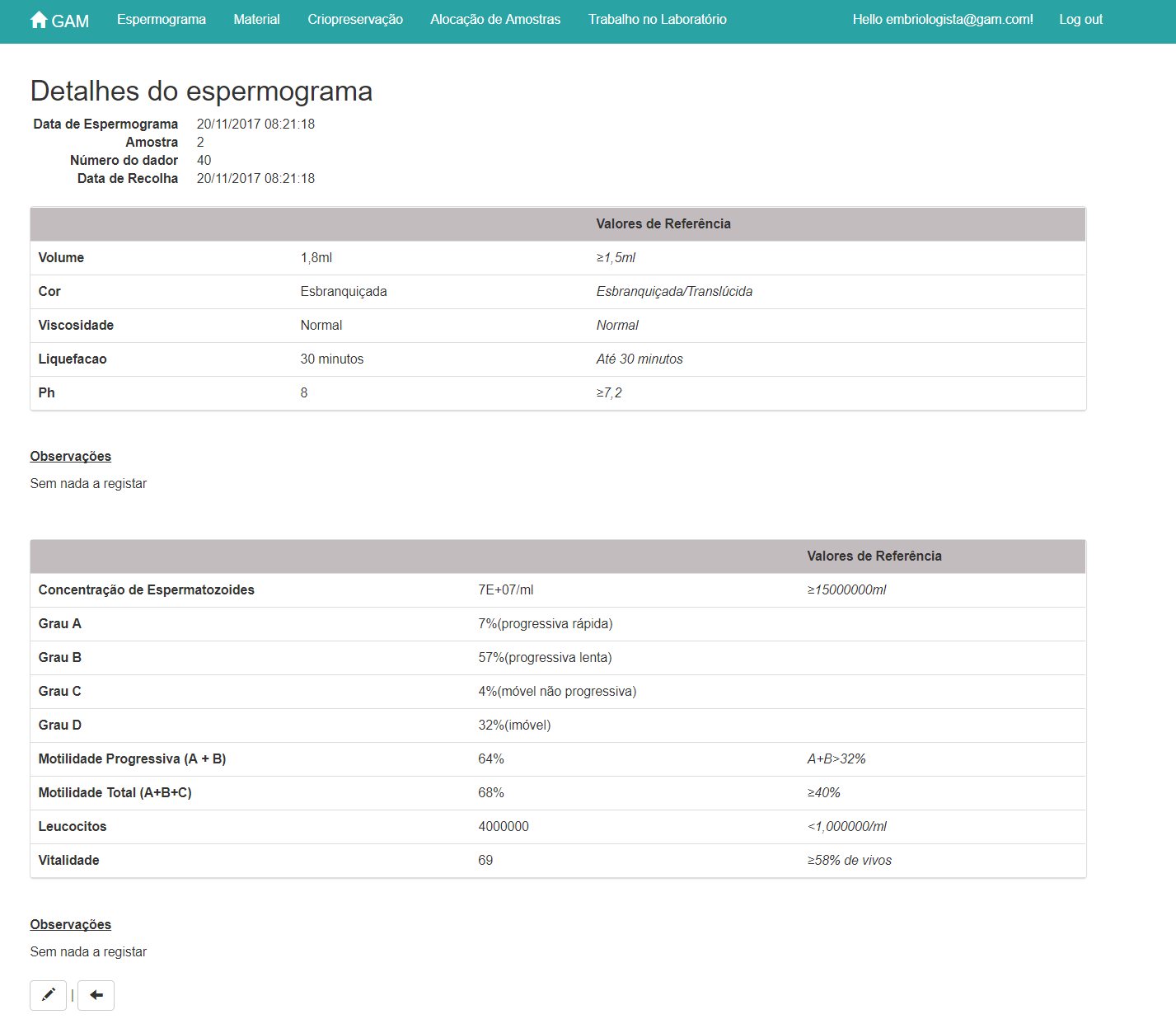


Figura 31 - REQ\_12: View dos detalhes do espermograma

## REQ 13 – Validação do espermograma

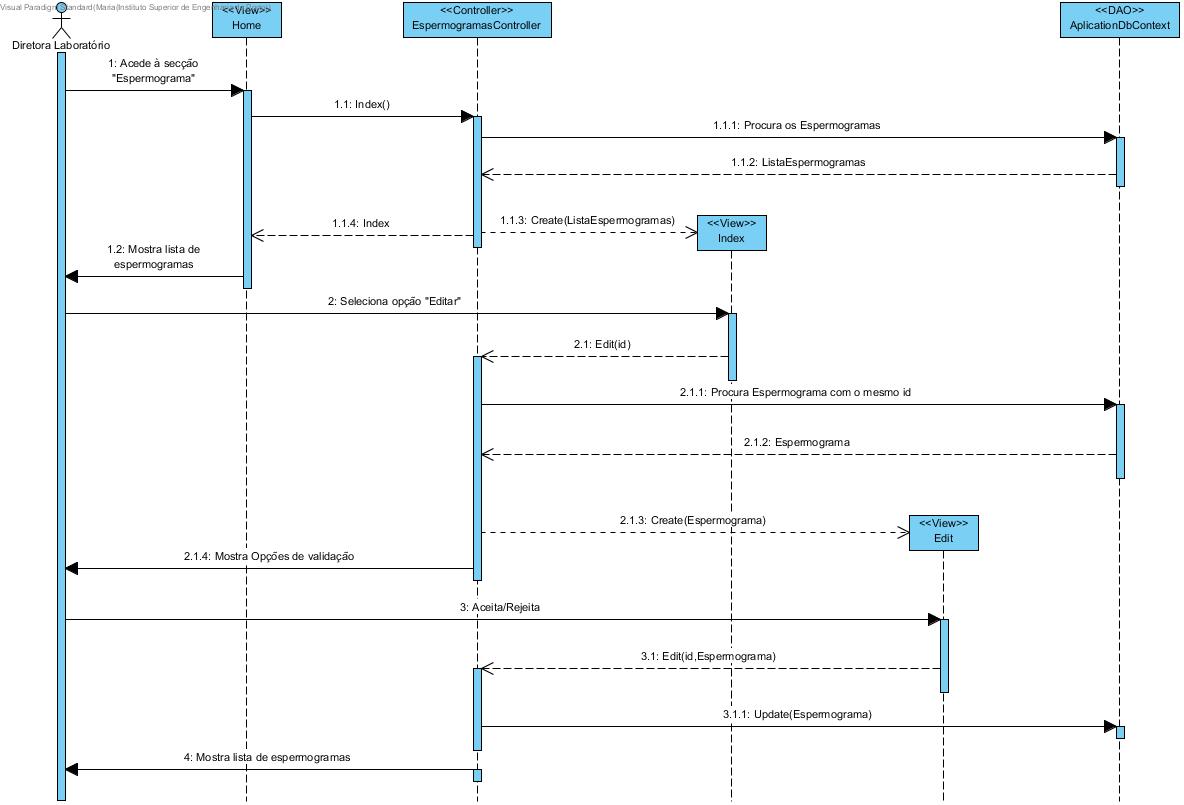
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 13 |
| Nome | Validação do espermograma |
| Descrição | A Diretora de Laboratório enquanto utilizadora registada no sistema, pode aceder à secção “Espermograma” na sua página da iGAM para validar espermogramas. Esta validação poderá ser aceite ou rejeitada. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – A Diretora de Laboratório acede à secção “Espermograma” | 1.1 – A view “Home” executa o método Index do Controller “EspermogramasController”  1.2 – O Controller “EspermogramasController” carrega a view de “Espermogramasr” associada, com espermogramas |
| 2 – A Diretora de Laboratório seleciona a opção “Editar” de um espermograma | 2.1 – A view “Espermogramas” executa o método Edit do Controller “EspermogramasController”  2.2 – O Controller “EspermogramasController” carrega a view de “Espermogramas” associada, com o Espermograma selecionado |
| 3 – A Diretora de Laboratório aceita ou rejeita o registo | 3.1 – A view “Espermograma” executa o método Edit do Controller “EspermogramasController”  3.2 - O Controller “EspermogramasController” atualiza o estado da validação do espermograma para aceite ou rejeitado conforme a escolha |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| A Diretora de Laboratório deve estar autenticada no sistema | O estado relativo à validação do espermograma é atualizado na base de dados | Sem nada a registar |

### Diagrama de sequência



### Ilustração da funcionalidade

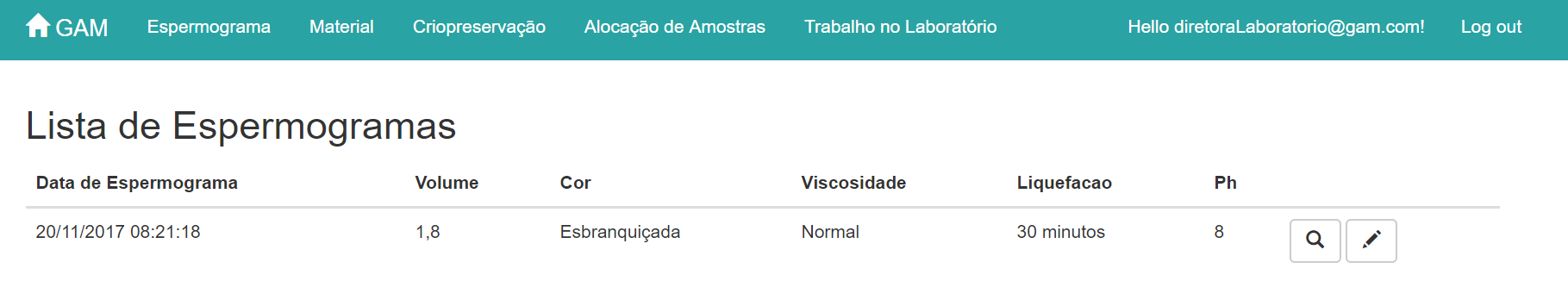


Figura 32 - REQ\_13: View da lista de espermogramas

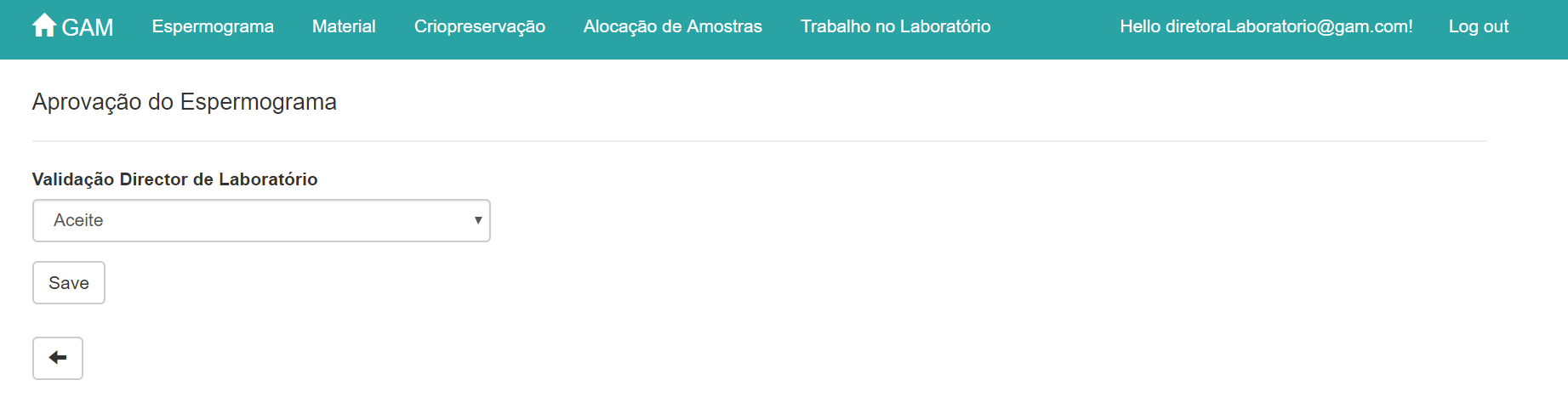


Figura 33 - REQ\_13: View de aprovação/rejeição de um espermograma

## REQ 14 – Criopreservação da amostra

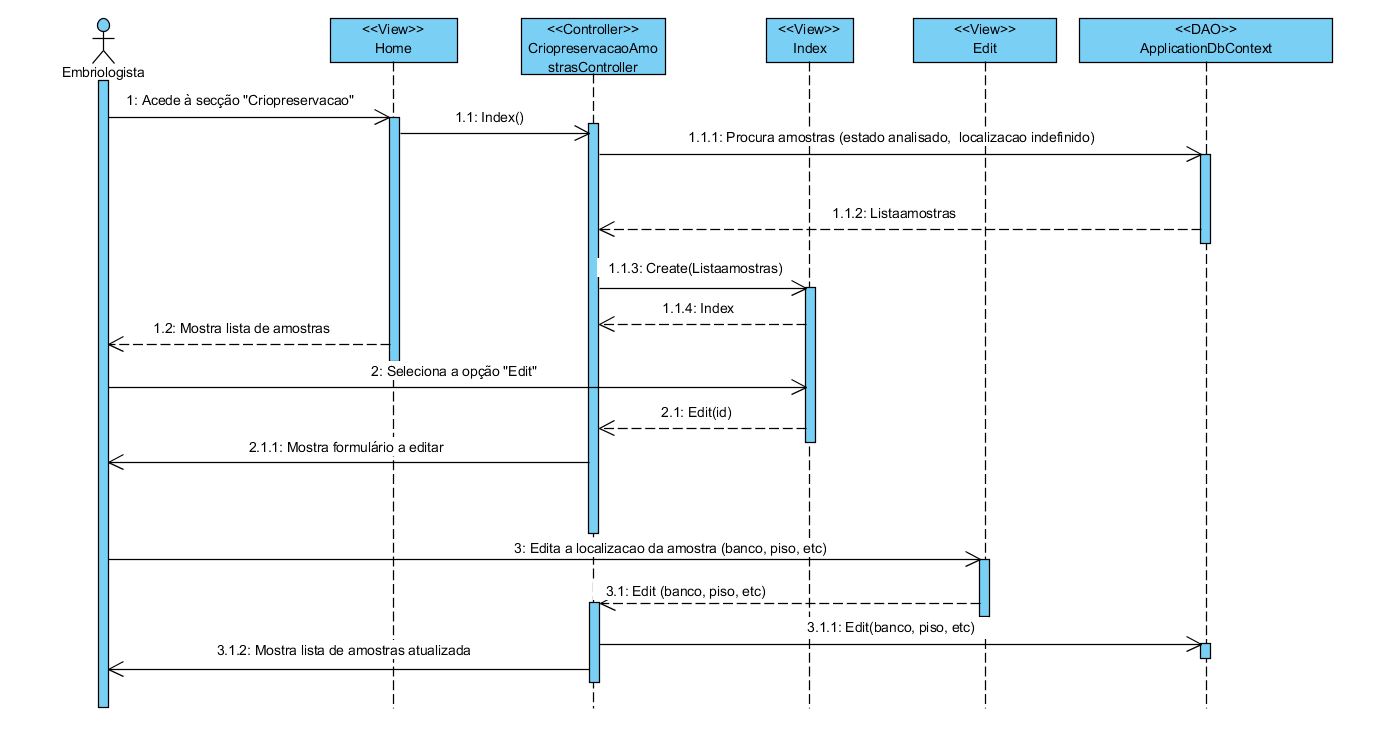
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 14 |
| Nome | Criopreservação da amostra |
| Descrição | A embriologista, enquanto utilizador registado no sistema, pode aceder à secção “Criopreservação” na sua página da iGAM para editar a localização da amostra criopreservada. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – A Embriologista acede à secção “Criopreservação” | 1.1 – A view “Home” executa o método Index do Controller “CriopreservacaoAmostrasController”  1.1. 2 – O Controller “CriopreservacaoAmostrasController” carrega a view “CriopreservacaoAmostras” associada, com as amostras analisadas do tipo “Esperma” com alocalização indefinido |
| 2 – A Embriologista escolhe a opção editar amostra | 2.1 – A view “CriopreservacaoAmostras” executa o método Edit do Controller “CriopreservacaoAmostrasController”  2.1.1 – O Controller “CriopreservacaoAmostrasController” carrega a view de “CriopreservacaoAmostras” associada, com um formulário para edição da amostra |
| 3 – A Embriologista atualiza a localização da amostra (banco, piso, cannister, goblet cor, goblet numero, palheta cor) | 3.1 - O Controller “CriopreservacaoAmostrasController” carrega a view de “CriopreservacaoAmostras” associada, com a lista de amostras atualizadas |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| A Embriologista deve estar autenticada no sistema | A localização da amostra é atualizada na base de dados | Sem nada a registar |
| A amostra deve estar no estado “analisada” e com localização “indefinido” |  |  |

### Diagrama de sequência



### Ilustração da funcionalidade

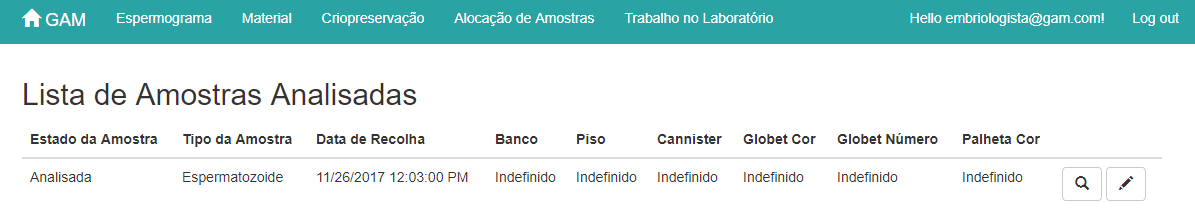


Figura 34 - REQ\_14: View da lista de amostras analisadas



Figura 35 - REQ\_14: View dos detalhes de uma amostra

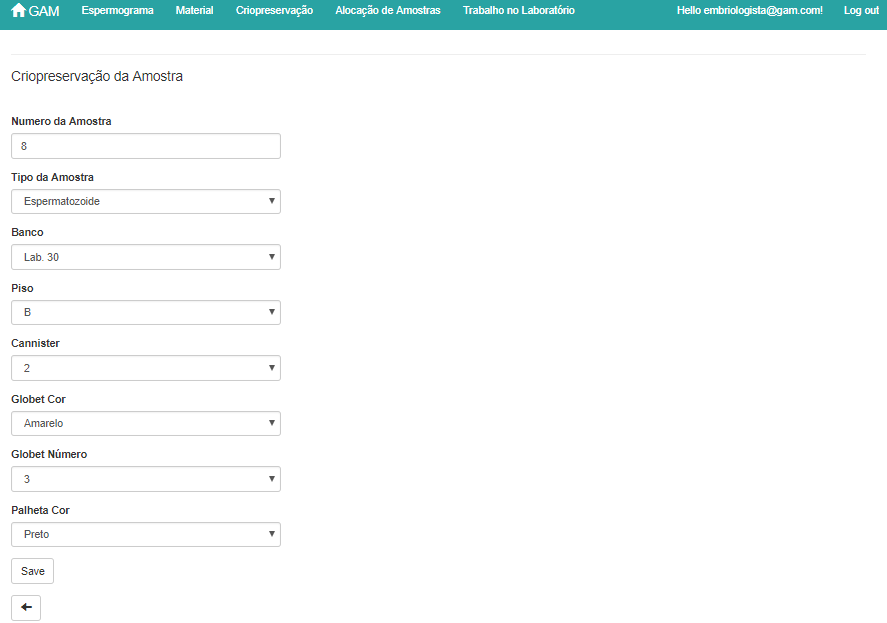


Figura 36 - REQ\_14: View da criopreservação de uma amostra

## REQ 15 – Visualização da alocação das amostras no Banco de criopreservação

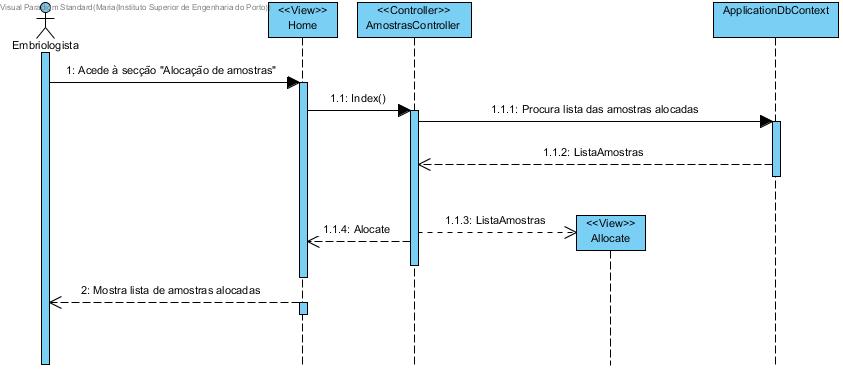
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 15 |
| Nome | Visualização da alocação das amostras no Banco de Criopreservação |
| Descrição | O Embriologista, enquanto utilizador registado do sistema, pode aceder à secção de “Alocação” na sua página da iGAM para visualizar a lista de amostras e as suas alocações. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – O Embriologista acede à secção “Alocação de amostras” | 1.1 – A view “Home” executa o método Index do Controller “AmostrasController” |
| 2 – O Embriologista visualiza a lista de amostras alocadas atualizada | 2.1 - O Controller “ValidacaoDadorController” carrega a view respectiva, com a lista de amostras e a sua alocação. |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| O Embriologista deve estar autenticado no sistema | Sem nada a registar | Sem nada a registar |

### Diagrama de sequência



### Ilustração da funcionalidade



Figura 37 - REQ\_15: View da lista de alocação das amostras

## REQ 17 – Registo de pedidos de gâmetas

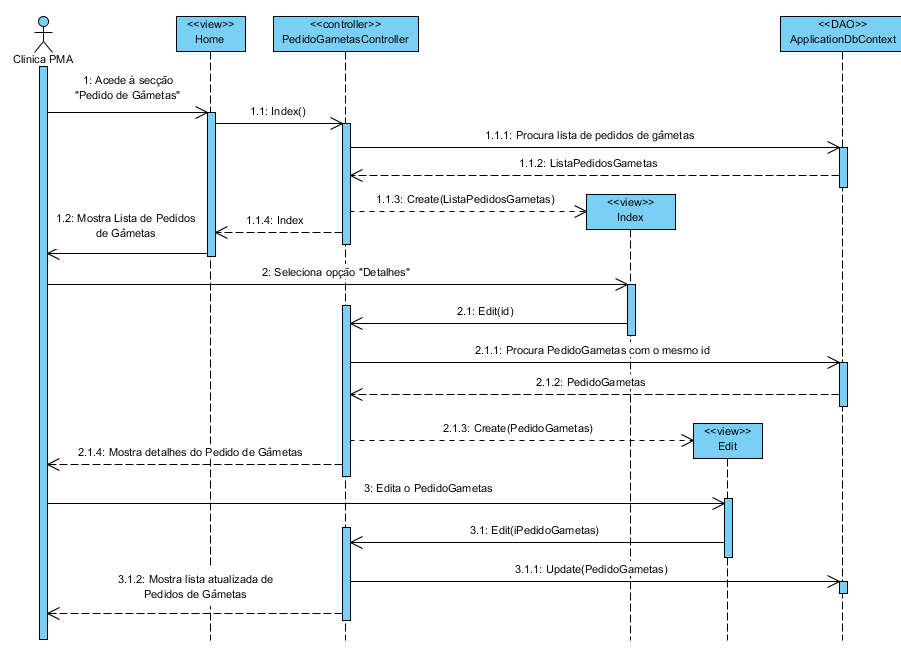
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 17 |
| Nome | Registo de pedido de gâmetas |
| Descrição | A Clínica PMA, enquanto utilizador registado do sistema, pode submeter um pedido de gâmetas na plataforma iGAM, preenchendo um formulário com os dados do casal interessado. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – A Clínica PMA acede à secção “Pedido de Gâmetas” | 1.1 – A view “Home” executa o método “Index” do Controller “PedidoGametasController”  1.2 – O Controller “PedidoGametasController” carrega a view de “PedidoGametas” associada, com os pedidos submetidos pela Clínica |
| 2 – A Clínica PMA escolhe uma das opções: “Criar novo pedido”,  “Ver detalhes”/“Editar”/“Apagar” um pedido já submetido | 2.1 – A view “PedidoGametas” executa um dos seguintes métodos: “Create”, “Details”, “Edit” ou “Delete”, pertencentes ao Controller “PedidoGametasController” |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| A Clínica PMA tem de ser um utilizador registado do sistema | Os pedidos submetidos devem ser aceites/rejeitados pela Enfermeira Coordenadora | Os campos de texto não podem estar vazios |
|  |  | Os campos numéricos não podem conter valores negativos |

### Diagrama de sequência



### Ilustração da funcionalidade

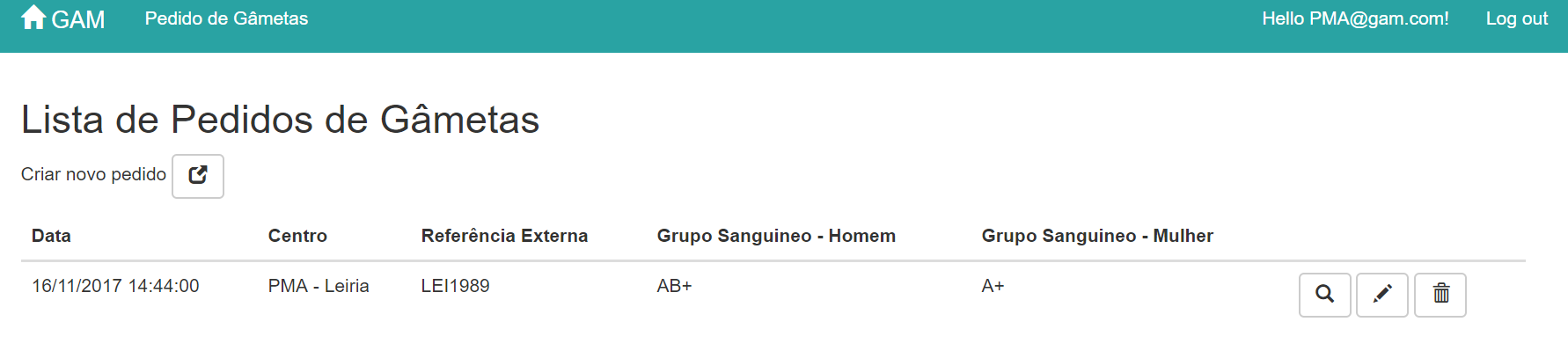


Figura 38 - REQ\_17: View da lista de pedidos de gâmetas

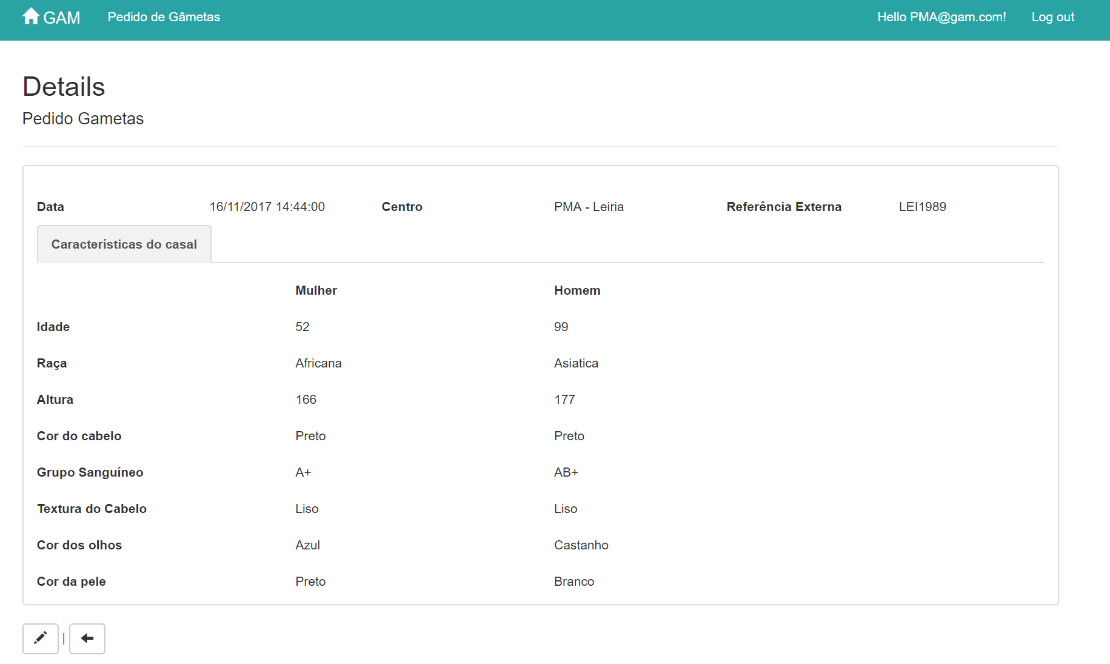


Figura 39 - REQ\_17: View de detalhes de um pedido de gâmetas

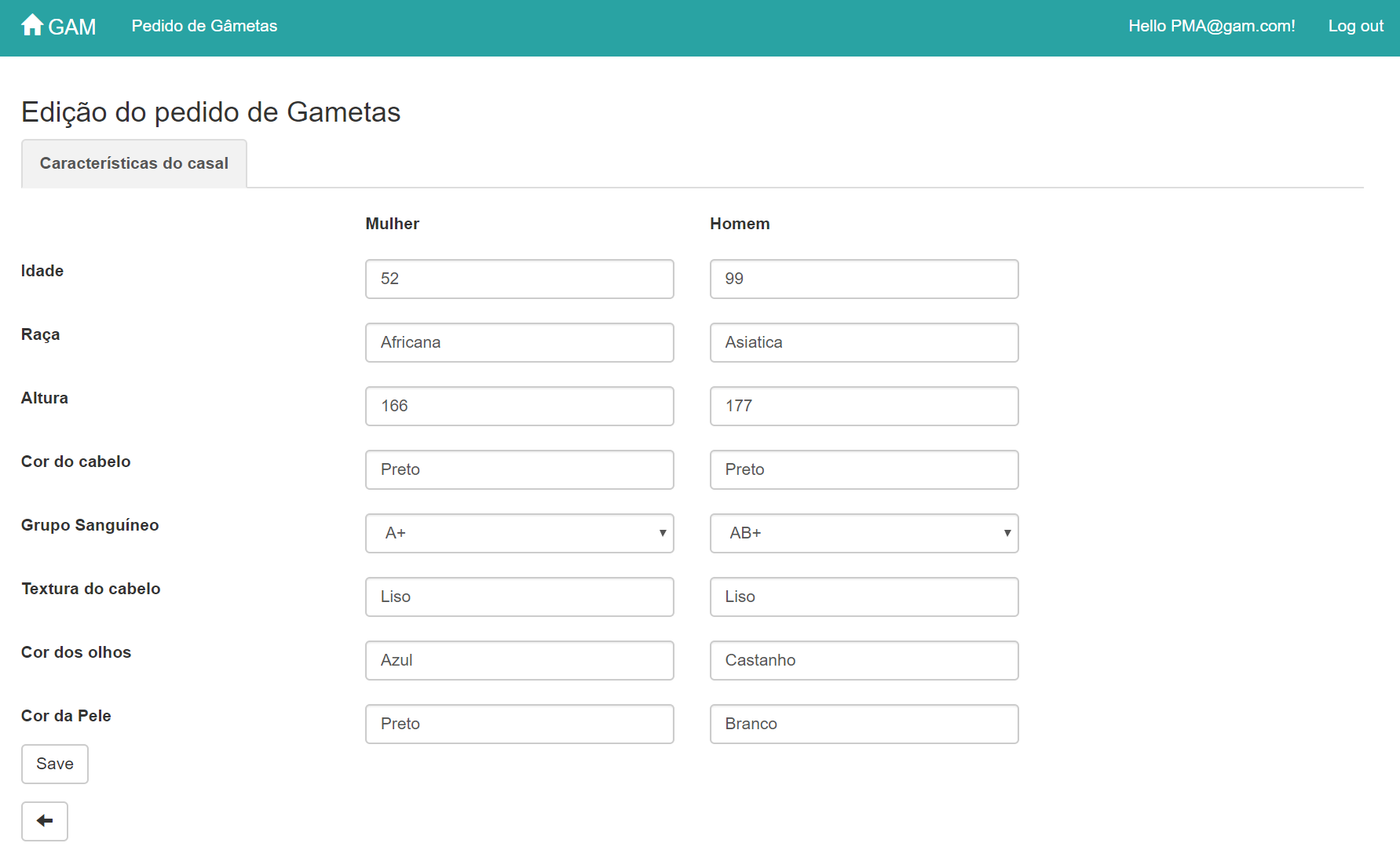


Figura 40 - REQ\_17: View de edição de um pedido de gâmetas

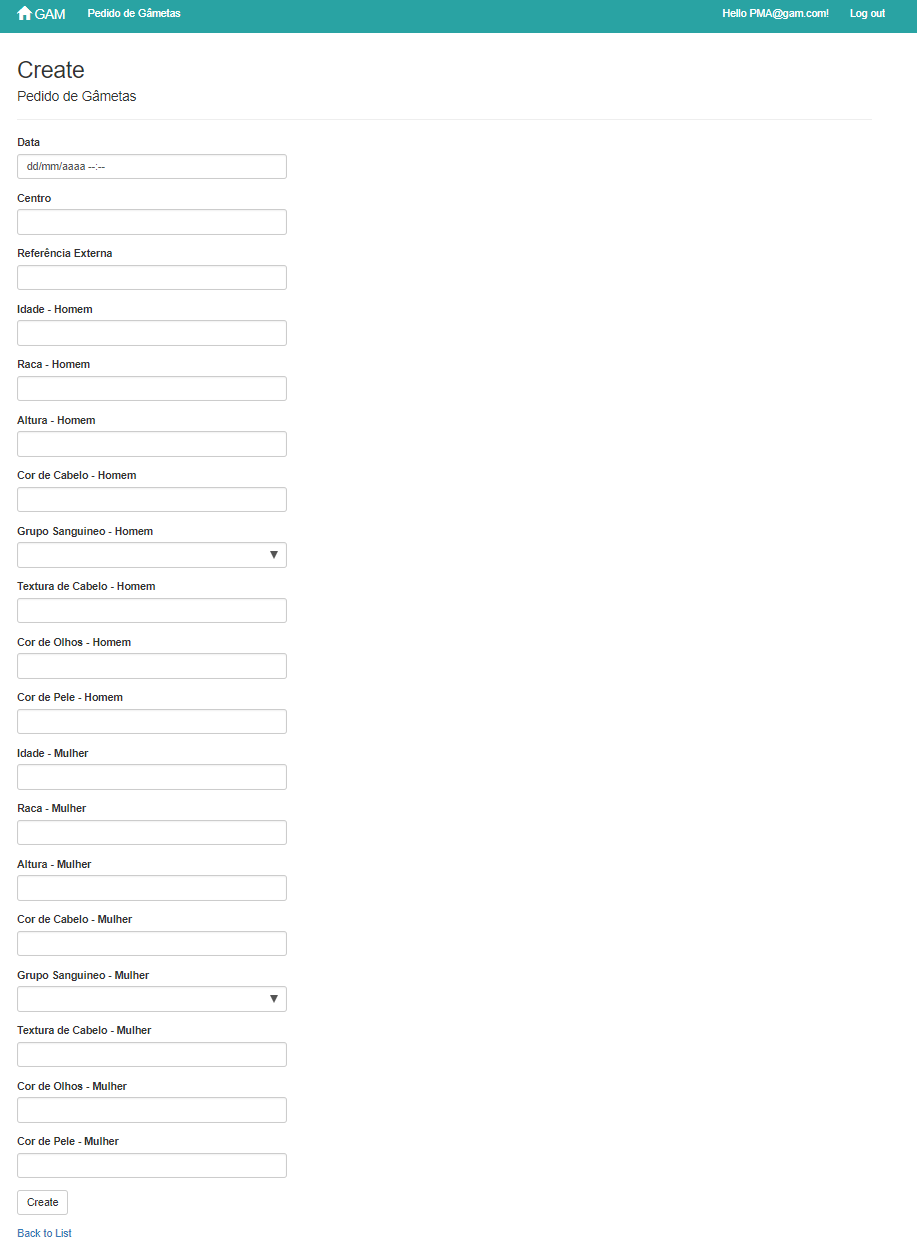


Figura 41 - REQ\_17: View da criação de um pedido de gâmetas

## REQ 18 – Registo de materiais usados (sem integração)

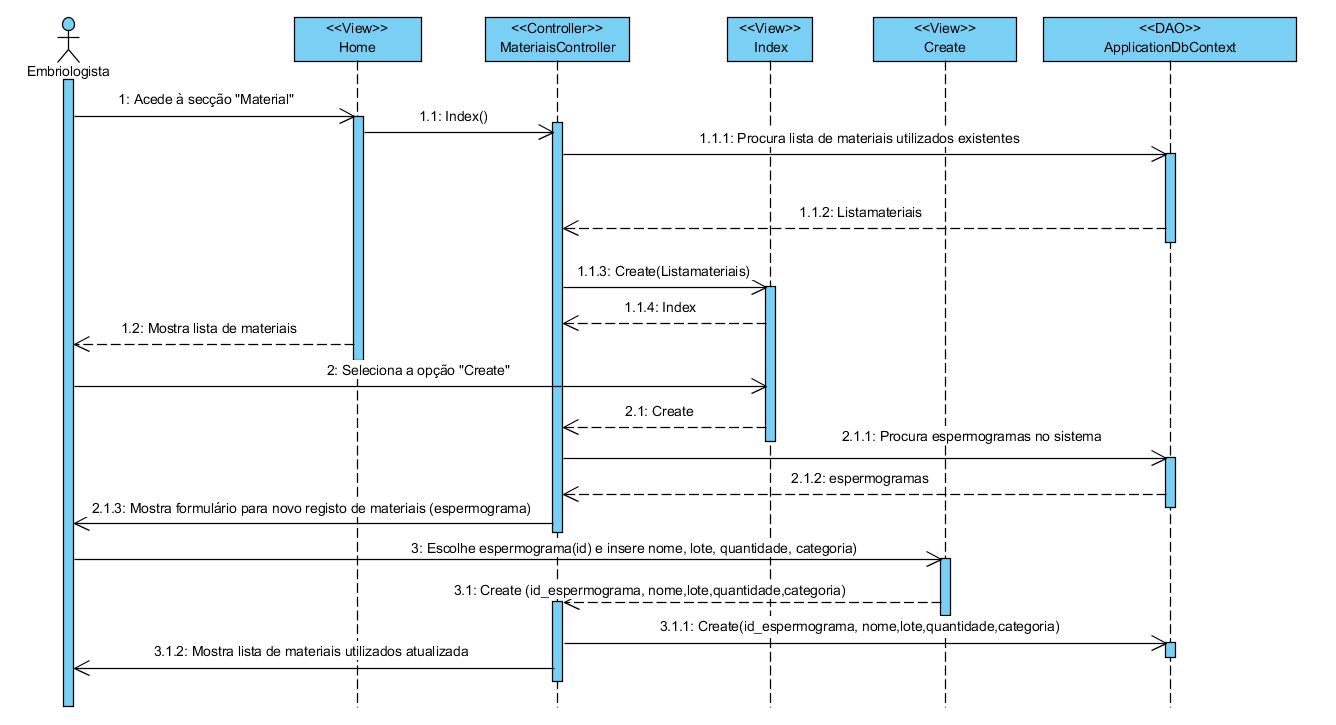
### Análise de requisitos

|  |  |
| --- | --- |
| Identificador | REQ 18 |
| Nome | Registo de materiais usados (sem integração) |
| Descrição | A embriologista, enquanto utilizador registado no sistema, pode aceder à secção “Material” na sua página da iGAM para criar, editar, ou eliminar os materiais utilizados. |

|  |  |
| --- | --- |
| Fluxo de Eventos - Humano | Fluxo de Eventos - Software |
| 1 – A Embriologista acede à secção “Material” | 1.1 – A view “Home” executa o método Index do Controller “MateriaisController”  1.1.2 – O Controller “MateriaisController” carrega a view de “Materiais” associada, com as consultas existentes no sistema |
| 2 – A Embriologista escolhe uma das opções: registar material\*, editar material ou remover material. | 2.1 – A view “Materiais” executa os métodos Create, Edit, ou Delete do Controller “MateriaisController” consoante a respetiva opção  2.1.1 – O Controller “MateriaisController” carrega a lista de espermogramas do sistema  2.1.3 – O Controller “MateriaisController” carrega a view de “Consultas” associada, com um formulário para criação ou edição do material, ou um pedido de confirmação para eliminação do material, consoante a respetiva opção |
| 3 – A Embriologista visualiza o formulário de registo de material (necessita de associar um espermograma, inserir nome, quantidade utilizada, lote e categoria) | 3.1 – A view “Create” executa o método Create do Controller “MateriaisController”  3.1.1 - O Controller “MateriaissController” executa o método Create(id\_espermograma,nome,quantidade,lote,categoria) da ApplicationDbContext.  3.1.2 – O sistema apresenta a lista de consultas atualizada |

|  |  |  |
| --- | --- | --- |
| Pré-condições | Pós-condições | Validações |
| A Embriologista deve estar autenticada no sistema | O material é registado/atualizado/removido da base de dados | Sem nada a registar |
| O espermograma deve estar criado no sistema |  |  |

### Diagrama de sequência



### Ilustração da funcionalidade

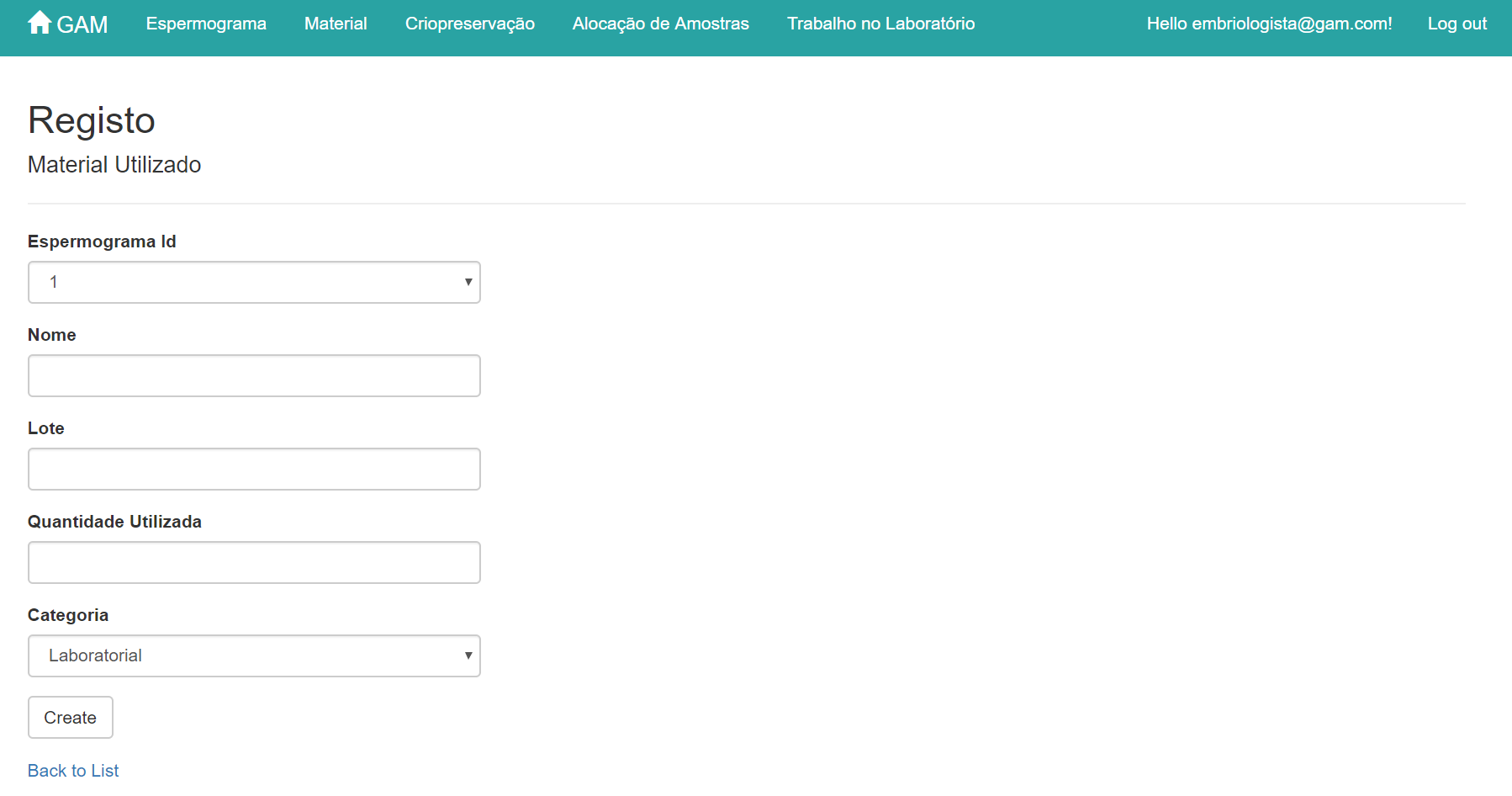


Figura 42 - REQ\_18: View do registo de material utilizado



Figura 43 - REQ\_18: View da lista de materiais usados

# Funcionalidades não implementadas

Todas as funcionalidades escalonadas para desenvolvimento neste *sprint* foram implementadas com sucesso. Contudo, a aprovação das mesmas ainda se encontra pendente da avaliação do público alvo.

# Pipeline

Para esta primeira iteração criamos um pipeline para o processo de desenvolvimento do *software*. Para a definição e implementação do pipeline, utilizamos o Jenkins, ferramenta de integração contínua e automatizada. O Jenkins auxilia na automatização do processo de desenvolvimento de *software* combinando os conceitos de *continuous integration* e *continuous delivery*.

A definição do script encontra-se no ficheiro *Jenkinsfile* na raiz do projeto. O script contém cinco *stages*:

* **Checkout**
  + Responsável por verificar a conexão ao repositório no Bitbucket, através de uma chave ssh

* **Build**
  + Responsável por compilar a solução GAM.sln no diretório correto
* **Unit Tests**
  + Responsável por compilar a solução GamTest.sln (testes unitários)
* **Publish Unit Test Results**
  + Responsável por criar um ficheiro de resultados (falhas) referente aos testes unitários (os ficheiros ficam guardados na pasta “TestResults”) e, deste modo, podemos analisar o número total de testes, testes que passaram/falharam, entre outros)

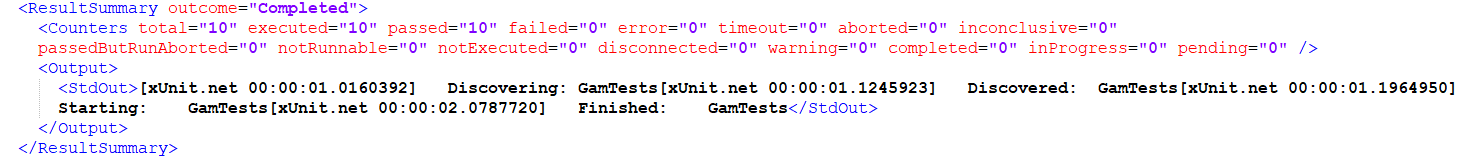


Figura 44 - Excerto do ficheiro de resultados (testes)

* **Archive**
  + Responsável por arquivar todos os artefactos

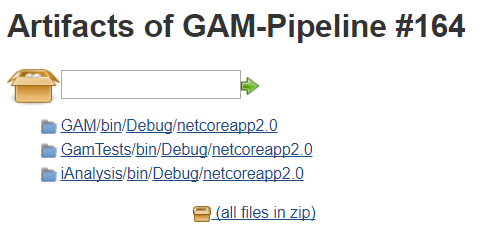


Figura 45 – Artefactos criados após stage Archive

É de salientar que o pipeline realiza *builds* periódicas de hora a hora. Para a implementação do pipeline foi necessário a instalação de alguns *plugins*, tais como: *Credentials Plugin, Git Plugin, MSBuild Plugin, Nuget Plugin, Pipeline, Pipeline: Stage View Plugin, MSTest Plugin e XUnit Plugin.*

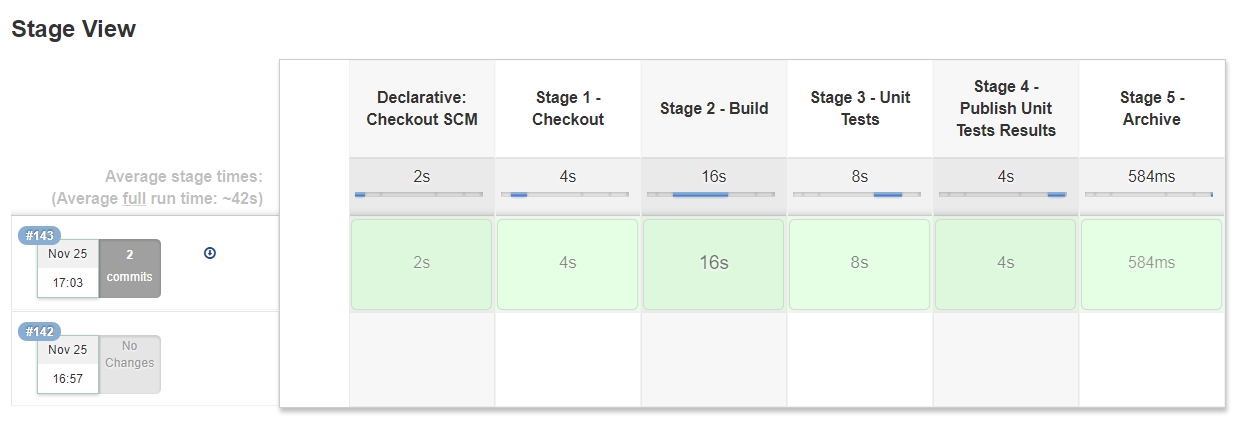


Figura 46 - Stage View Pipeline

# Distribuição de tarefas

A distribuição de tarefas foi realizada através de um *board*, usando a ferramenta Trello. Esta ferramenta é muito flexível, auxilia na criação de tarefas podendo a mesma ser atribuída a um membro da equipa e passar por diferentes estados (To Do, Doing, Done). Os requisitos são denominados *cards*, e podem ser deslocados consoante o estado em que se encontram.

Na tabela 2 apresentamos os requisitos que foram atribuídos a cada membro da equipa. É de salientar que o requisito 1 foi desenvolvido por toda a equipa.

Tabela 2 – Distribuição tarefas

|  |  |  |
| --- | --- | --- |
| Id | Identificação | Membro |
| REQ\_1 | Registo inicial do dador (sem fotografia) | Todos |
| REQ\_2 | Validação dos dados do dador | Tiago Gonçalves |
| REQ\_3 | Gestão de perfis de utilizadores | Tiago Gonçalves |
| REQ\_4 | Configuração do template de perguntas médicas | Daniel Bento |
| REQ\_5 | Consultas médicas via inquérito template | Daniel Bento |
| REQ\_6 | Registo de amostras | Maria Almeida |
| REQ\_7 | Pedido das análises de sangue iniciais | Manuel Correia |
| REQ\_8 | Ciclo da dádiva (abertura e consulta de informação) | Tiago Gonçalves |
| REQ\_9 | Marcação de consultas | Ana Barros |
| REQ\_10 | Aprovação / Rejeição do ciclo da dádiva | Tiago Gonçalves |
| REQ\_11 | Lista de trabalho do laboratório | Daniel Bento |
| REQ\_12 | Registo de espermograma | Maria Almeida |
| REQ\_13 | Validação do espermograma | Maria Almeida |
| REQ\_14 | Criopreservação da amostra | Ana Barros |
| REQ\_15 | Visualização da alocação das amostras no Banco de criopreservação | Maria Almeida |
| REQ\_16 | Integração com o laboratório de análises de sangue | Manuel Correia |
| REQ\_17 | Registo de pedidos de gâmetas | Manuel Correia |
| REQ\_18 | Registo de materiais usados (sem integração) | Ana Barros |
|  | Definição do pipeline | Ana Barros |

# Tempos

Tabela 3 - Duração de cada requisito

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Identificação | Membro | Duração |
| REQ\_1 | Registo inicial do dador (sem fotografia) | Todos | 9h |
| REQ\_2 | Validação dos dados do dador | Tiago Gonçalves | 1h |
| REQ\_3 | Gestão de perfis de utilizadores | Tiago Gonçalves | 8h |
| REQ\_4 | Configuração do template de perguntas médicas | Daniel Bento | 8h |
| REQ\_5 | Consultas médicas via inquérito template | Daniel Bento | 4h |
| REQ\_6 | Registo de amostras | Maria Almeida | 3:30h |
| REQ\_7 | Pedido das análises de sangue iniciais | Manuel Correia | 4h |
| REQ\_8 | Ciclo da dádiva (abertura e consulta de informação) | Tiago Gonçalves | 3h |
| REQ\_9 | Marcação de consultas | Ana Barros | 2h |
| REQ\_10 | Aprovação / Rejeição do ciclo da dádiva | Tiago Gonçalves | 2h |
| REQ\_11 | Lista de trabalho do laboratório | Daniel Bento | 2:30h |
| REQ\_12 | Registo de espermograma | Maria Almeida | 2:30h |
| REQ\_13 | Validação do espermograma | Maria Almeida | 3h |
| REQ\_14 | Criopreservação da amostra | Ana Barros | 4h |
| REQ\_15 | Visualização da alocação das amostras no Banco de criopreservação | Maria Almeida | 3:30h |
| REQ\_16 | Integração com o laboratório de análises de sangue | Manuel Correia | 7:30h |
| REQ\_17 | Registo de pedidos de gâmetas | Manuel Correia | 4h |
| REQ\_18 | Registo de materiais usados (sem integração) | Ana Barros | 2h |
|  | Definição do pipeline | Ana Barros | 7h |

Quanto ao tempo que demorou cada requisito a ficar concluído, importa referir que se trata de uma estimativa uma vez que não foi utilizada nenhuma ferramenta para a medição real do tempo. Este é porventura um dos pontos sobre os quais devemos refletir e tentar melhor no próximo *sprint*. De realçar que o desenvolvimento do trabalho destinado a este *sprint* foi sempre pautado por uma grande entreajuda dos elementos da equipa de desenvolvimento, tendo sido constante a colaboração de vários elementos para a realização de diversos *issues*. Por fim, importa referir que, como é natural, foram existindo variadas tarefas ao longo do processo de desenvolvimento (como por exemplo a resolução de *bugs*), que ocuparam tempo considerável aos elementos do grupo e que não foi possível quantificar e associar à tabela apresentada anteriormente. Os valores mencionados na tabela 3 perfazem um total de 100 horas de trabalho de desenvolvimento.

# Medidas de qualidade de software

Quanto a medidas de qualidade de software, a equipa de desenvolvimento tentou sempre manter o código limpo e isolado por camadas, de modo a construir uma estrutura organizada e escalável. No mesmo setor da escalabilidade encontra-se a nossa decisão de migrar a base de dados do projeto para a *cloud*. Por fim, desenvolvemos alguns testes unitários às funcionalidades da aplicação. Contudo, o nível de profundidade dos mesmos não está ainda nos parâmetros que consideramos necessários para garantir uma aplicação altamente fiável, pelo que este é, sem sombra de dúvida, um dos pontos sobre o qual o grupo de trabalho vai refletir de modo a tentar melhorar na próxima iteração.