**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** |

2017

**Índice**

[**LABDSOFT** 0](#_Toc499393297)

[**ISEP – Mestrado em Engenharia Informática** 0](#_Toc499393298)

[**Banco de Gâmetas - GAM** 0](#_Toc499393299)

[Trabalho prático nr.2 – iteração 1 0](#_Toc499393300)

[**Histórico de Revisão** 0](#_Toc499393301)

[**Índice** 1](#_Toc499393302)

[**Índice de Ilustrações** 2](#_Toc499393303)

[**Índice de Tabelas** 2](#_Toc499393304)

[1 Introdução 3](#_Toc499393305)

[1.1 Propósito 3](#_Toc499393306)

[1.2 Scope do produto 3](#_Toc499393307)

[1.3 Funcionalidades propostas 3](#_Toc499393308)

[2 Decisões tomadas 5](#_Toc499393309)

[3 Artefactos 6](#_Toc499393310)

[3.1 Diagrama de contexto 6](#_Toc499393311)

[3.2 Modelo de dados 6](#_Toc499393312)

[3.3 Diagrama de arquitetura 7](#_Toc499393313)

[4 Pipeline 8](#_Toc499393314)

[5 Funcionalidades implementadas 10](#_Toc499393315)

[6 Funcionalidades não implementadas 11](#_Toc499393316)

[7 Distribuição de tarefas 12](#_Toc499393317)

[8 Tempos 13](#_Toc499393318)

[9 Medidas de qualidade de software 14](#_Toc499393319)

**Índice de Ilustrações**

[Figura 1 - Diagrama de Contexto 6](file:///C:\Users\Altran\Desktop\LABDSOFT\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499393353)

[Figura 2 - Modelo de dados 6](file:///C:\Users\Altran\Desktop\LABDSOFT\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499393354)

[Figura 3 - Diagrama de arquitetura 7](file:///C:\Users\Altran\Desktop\LABDSOFT\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499393355)

[Figura 4 – Artefactos criados após stage Archive 8](file:///C:\Users\Altran\Desktop\LABDSOFT\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499393356)

[Figura 5 - Excerto do ficheiro de resultados (testes) 9](file:///C:\Users\Altran\Desktop\LABDSOFT\Documentacao\Entrega%202\relatorio_tp2_g1.docx#_Toc499393357)

**Índice de Tabelas**

Tabela 1 - Funcionalidades propostas 4

Tabela 2 – Distribuição tarefas 12

# 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* da Microsoft Azure. 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

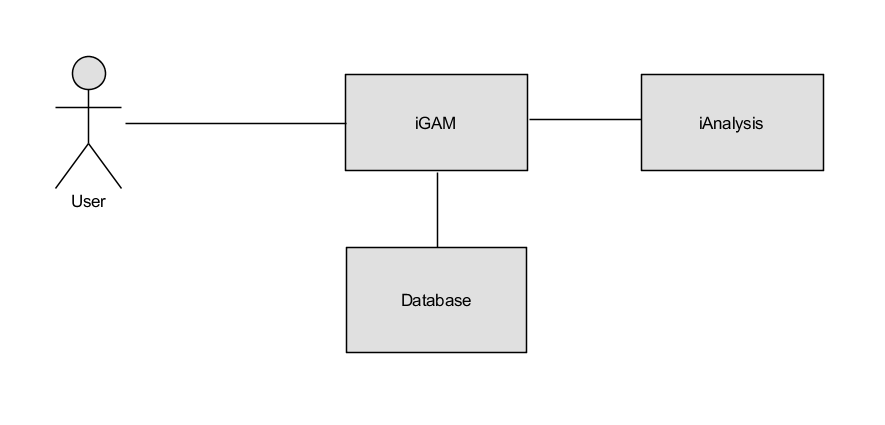


Figura 1 - Diagrama de Contexto

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 plataforma Microsoft Azure, 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

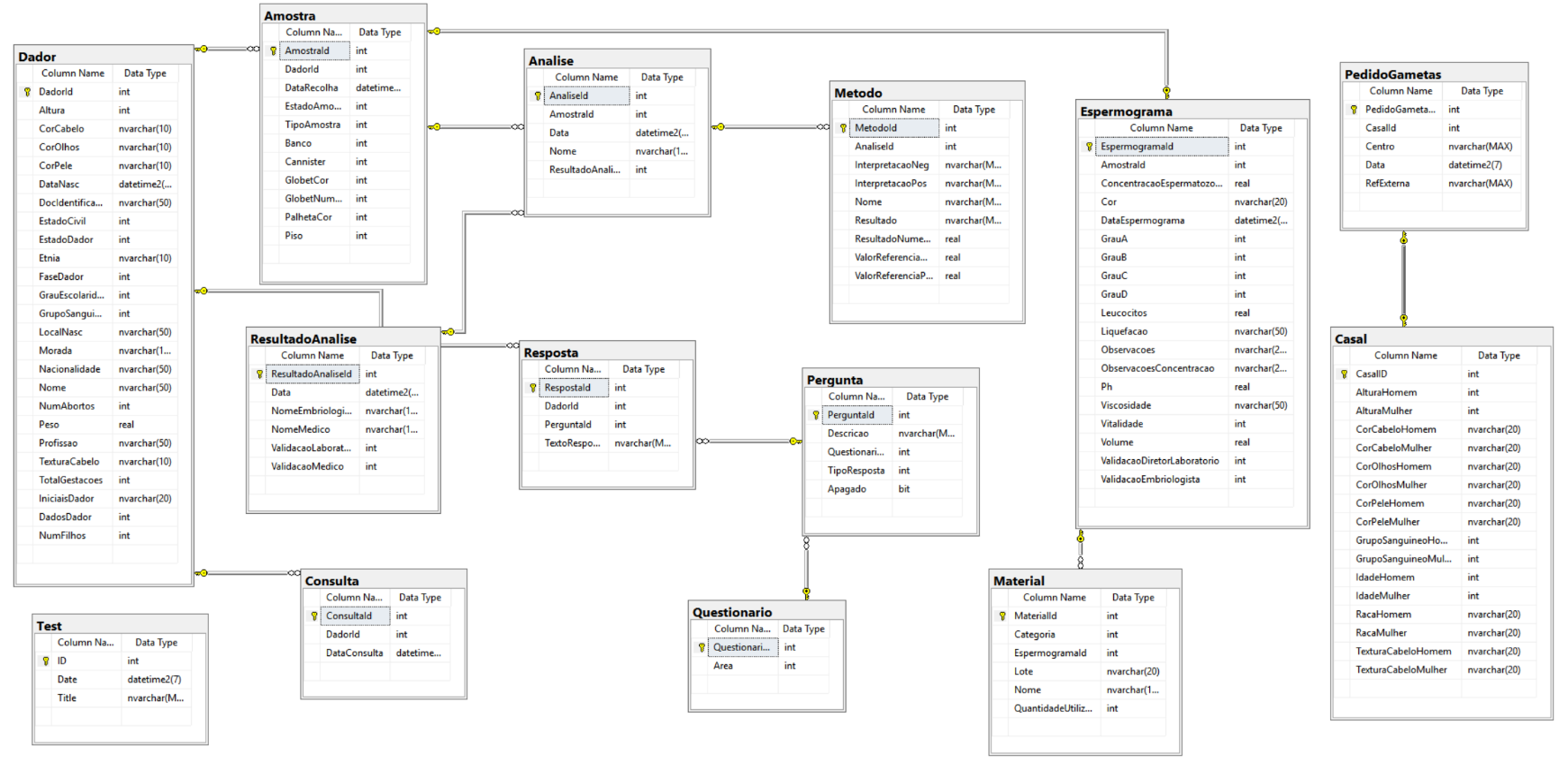


Figura 2 - Modelo de dados

## Diagrama de arquitetura

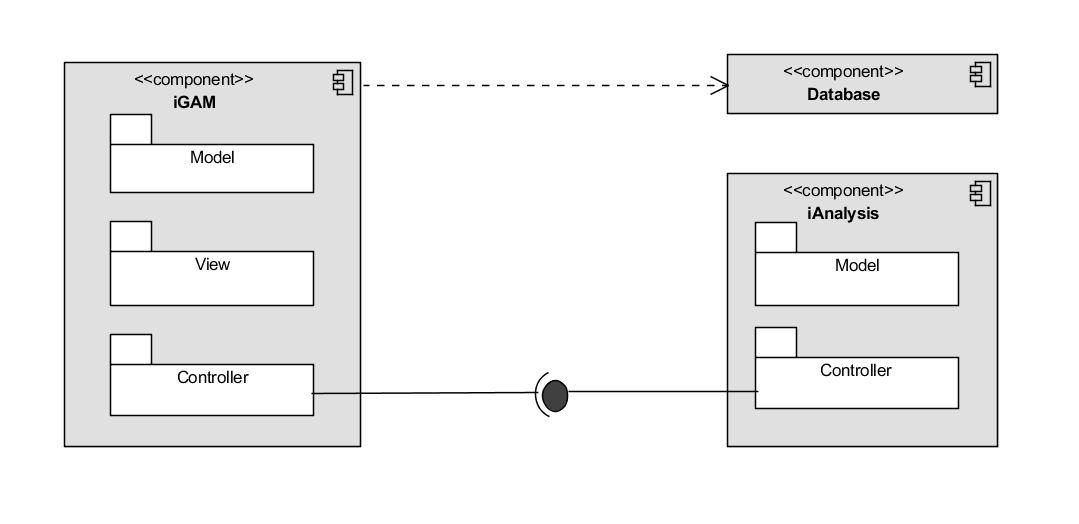


Figura 3 - Diagrama de arquitetura

# 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)

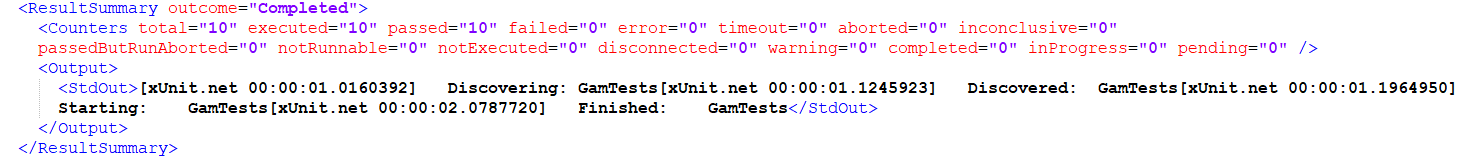


Figura 4 - Excerto do ficheiro de resultados (testes)

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

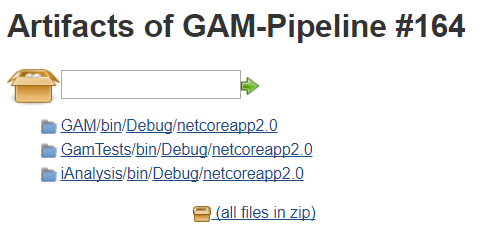


Figura 5 – Artefactos criados após stage Archive

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.*

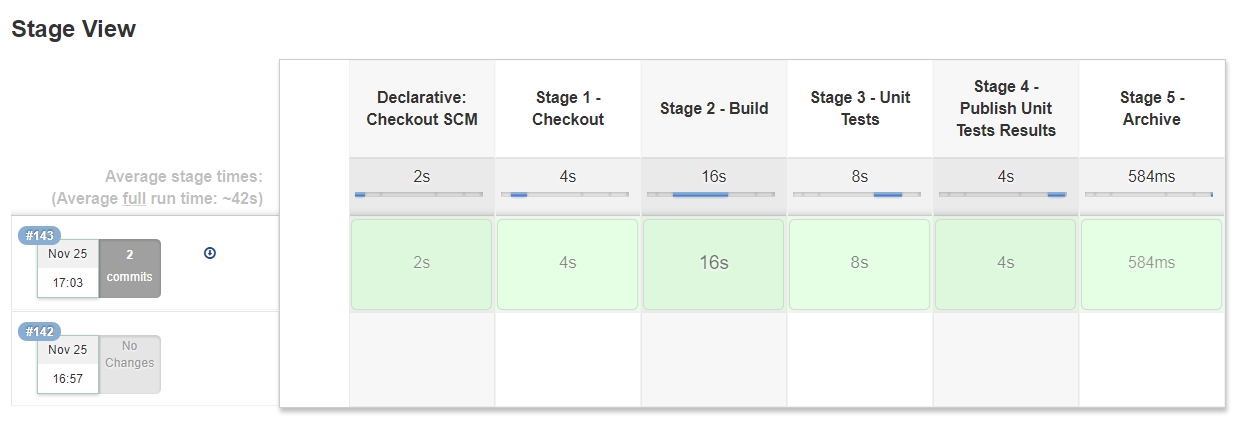


Figura 6 - Stage View Pipeline

# Funcionalidades implementadas

## REQ 1

## REQ 2

## REQ 3

## REQ 4

## REQ 5

## REQ 6

## REQ 7

## REQ 8

## REQ 9

## REQ 10

## REQ 11

## REQ 12

## REQ 13

## REQ 14

## REQ 15

## REQ 16

## REQ 17

## REQ 18

# 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.

# 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.

# Medidas de qualidade de software

Não aplicável (iteração 2 apenas).