

## **Planeamento e Gestão de Projeto**

Ano Letivo 2019/2020

### **Gestão De Grupos De Alunos Para Trabalhos Práticos A Nível Universitário**

#### **Relatório da Etapa 2**

##### **Grupo 09**

André Silva	51634
António Pinto	52045
Diogo Frazão	51595
Guilherme Nunes	51594
Tiago Robalo	51628
Vasco Bento	51636

# Índice

Introdução .....	4
1. Requisitos funcionais.....	5
1.1. Alunos:.....	5
1.2. Professores .....	5
1.3. Administradores do sistema.....	6
2. Requisitos não funcionais.....	7
2.1. Persistência e Disponibilidade.....	7
2.2. Segurança (Integridade e Confidencialidade) .....	7
2.3. Baixo custo e Escalabilidade.....	7
2.4. Usabilidade .....	7
2.5. Fiabilidade .....	7
3. Dados de entrada e saída .....	8
3.1. Dados de entrada .....	8
3.2. Dados de saída .....	9
4. Recursos.....	10
4.1. Pessoas da equipa .....	10
4.2. <i>Software</i> para o sistema.....	16
4.3. Ferramentas de desenvolvimento .....	16
4.3.1. Hardware.....	16
4.3.2. Software .....	16
4.3.3. Linguagens de Programação .....	17
4.3.4. Bibliotecas Externas .....	17
4.3.5. Frameworks.....	17
4.3.6. Serviços de Comunicação.....	17
5. Estimativas.....	19
5.1. Esforço disponível .....	19
5.2. Dados históricos .....	20
5.3. Estimativas baseadas em linhas de código .....	27

5.3.1. Requisitos Funcionais .....	27
5.3.2. Requisitos não funcionais .....	32
5.4. Estimativas baseadas no modelo COCOMO $E_{otimista}$ .....	35
5.5. Análise Crítica .....	37
6. Processo de desenvolvimento de <i>software</i> .....	38
7. Organização da equipa .....	40
8. Planeamento do projeto.....	42
8.1. <i>Work packages</i> .....	42
8.2. Calendarização .....	46
9. Gestão de riscos.....	48
9.1. Lista de riscos .....	48
9.2. Tabela de riscos .....	49
9.3. Plano RMMM .....	50
Conclusão.....	51
Bibliografia.....	54

## Introdução

No âmbito da unidade curricular, e no seguimento de Conceção de Produto, foi nos apresentada uma questão que envolvia a gestão de grupos de alunos para trabalhos práticos a nível universitário. Iremos dar seguimento ao desenvolvimento do tema nas cadeiras de Projeto de Tecnologias de Informação e de Projeto de Tecnologias de Redes pondo em prática as conclusões recolhidas no estudo.

O objetivo deste projeto é planejar o desenvolvimento de um sistema para suportar trabalhos em grupo em ambiente universitário. Este problema apresenta relativa importância porque ao longo da vida académica, todos os alunos irão ser confrontados com muitos desafios associados a trabalhos realizados em grupo. Estes desafios foram identificados em Conceção de Produto.

O sistema deve ser acessível através de uma interface web. Além disso, seria ideal a utilização do sistema em larga escala e que fosse possível a integração com sistemas existentes na Faculdade ou a melhoria dos mesmos, especialmente se apontarem para soluções inovadoras e criativas.

Os administradores do sistema devem poder assegurar a existência de dados, em cada ano letivo, sobre os alunos, as respetivas turmas e as unidades curriculares em que estão inscritos.

Os professores devem poder: definir as regras para a constituição de grupos para os projetos das suas unidades curriculares; estipular etapas para os projetos, com objetivos, entregáveis e prazos; consultar os documentos e outros artefactos que vão sendo produzidos por cada grupo em cada projeto; responder a dúvidas dos grupos; e dar feedback sobre o que cada grupo entrega em cada etapa.

Os alunos devem poder: criar grupos ou juntar-se a existentes; comunicar entre si ou com o professor, por exemplo para colocar dúvidas; guardar documentos e outros artefactos numa área de trabalho do grupo; combinar períodos em que estão disponíveis para reunir; registar o tempo que dedicam a cada tarefa do projeto; e atribuir pontuações às prestações dos colegas, para guiar futuras constituições de grupos.

A solução deverá assegurar a disponibilidade, correção, e confidencialidade dos dados, através da tolerância a falhas por parte do sistema, que deve também ser escalável e assegurar que são cumpridas as boas práticas para a manutenção da confidencialidade de dados.

Esperamos que com o desenvolvimento do nosso projeto, os alunos consigam usufruir de um sistema mais eficiente, hábil e capacitado de forma a responder de uma forma completa e eficaz às suas necessidades face ao processo de criação e gestão de grupos de alunos para trabalhos práticos no âmbito universitário.

## PARTE I — ÂMBITO DO PROJETO

### 1. Requisitos funcionais

#### 1.1. Alunos:

- RF-1      Efetuar login de utilizador
- RF-2      Efetuar logout de utilizador
- RF-3      Visualizar perfil de utilizador
- RF-4      Editar perfil do utilizador
- RF-5      Criar grupo de trabalho para a unidade curricular
- RF-6      Sair de um grupo de trabalho de uma unidade curricular
- RF-7      Pedir para juntar a grupo existente
- RF-8      Gerir pedidos para juntar a grupo
- RF-9      Comunicar entre elementos do grupo
- RF-10     Comunicar com outros alunos
- RF-11     Comunicar com o professor
- RF-12     Gerir documentos na área de trabalho do grupo
- RF-13     Submeter versão final dos trabalhos
- RF-14     Criar tarefas referentes a um trabalho
- RF-15     Terminar tarefas referentes a um trabalho
- RF-16     Gerir tarefas referentes a um trabalho
- RF-17     Inserir horário de disponibilidade
- RF-18     Ver horário de disponibilidade dos elementos do grupo
- RF-19     Atribuir pontuações/avaliação ao trabalho dos restantes elementos do grupo

#### 1.2. Professores

- RF-20     Efetuar login de docente
- RF-21     Efetuar logout de docente
- RF-22     Definir regras para constituição de grupos
- RF-23     Definir número de grupos possíveis
- RF-24     Publicar anúncios com informações referentes ao trabalho
- RF-25     Estipular etapas para os projetos (enunciado do projeto, objetivos, entregáveis e prazos)
- RF-26     Consultar os documentos e outros artefactos que vão sendo produzidos por cada grupo em cada projeto
- RF-27     Responder a dúvidas dos grupos
- RF-28     Comunicar com os alunos
- RF-29     Comunicar com outros professores
- RF-30     Dar feedback a cada grupo em cada etapa

### **1.3. Administradores do sistema**

- RF-31 Assegurar a existência de dados dos professores
- RF-32 Assegurar a existência de dados dos alunos
- RF-33 Criação de novas unidades curriculares
- RF-34 Assegurar a existência de dados das unidades curriculares
- RF-35 Criação de novas turmas
- RF-36 Assegurar a existência de dados das turmas
- RF-37 Importação de dados em ficheiros ou outras fontes
- RF-38 Assumir permissões distintas com base no perfil de utilizador
- RF-39 Retirar permissões

## **2. Requisitos não funcionais**

### **2.1. Persistência e Disponibilidade**

- RNF-1      Replicação de dados
- RNF-2      Disponível 24 horas por dia

### **2.2. Segurança (Integridade e Confidencialidade)**

- RNF-3      Canais seguros (cifras)
- RNF-4      Transações seguras
- RNF-5      Privacidade dos dados dos utilizadores
- RNF-6      Não é permitido o acesso não autorizado ao sistema
- RNF-7      Tolerar a falha de qualquer componente de hardware com redução mínima de desempenho e sem perda de dados;

### **2.3. Baixo custo e Escalabilidade**

- RNF-8      Cloud computing (AWS)
- RNF-9      O sistema deve ser escalável e modular, por forma a suportar facilmente a adição e remoção de hardware.

### **2.4. Usabilidade**

- RNF-10     Interface simples e intuitiva
- RNF-11     A interface suporta diferentes tipos de dispositivos, como por exemplo desktop, tablets e smartphones.

### **2.5. Fiabilidade**

- RNF-12     Tolerar a falha de uma qualquer componente de hardware com uma redução mínima de desempenho e sem perda de dados.
- RNF-13     Tolerar falhas catastróficas com duração não superior a um dia, sendo apenas admissível a perda de dados registados nessas últimas 24 horas.

### **3. Dados de entrada e saída**

#### **3.1. Dados de entrada**

##### **1. Administradores do Sistema:**

- a) Dados dos Alunos:
  - i) Nome
  - ii) Número (atributo que o caracteriza)
  - iii) E-mail
  - iv) Password
  - v) Curso
  - vi) Disponibilidade (horário)
- b) Dados dos professores:
  - i) Nome
  - ii) Número (atributo que o caracteriza)
  - iii) E-mail
  - iv) Password
- c) Unidades Curriculares:
  - i) Professores
  - ii) Alunos
  - iii) Código (atributo que o caracteriza)

##### **2. Professores:**

- d) Regras de constituição de Grupo
  - i) Número de elementos
  - ii) Pertencentes à mesma turma prática
- e) Definição da quantidade de grupos do projeto
  - i) Número máximo de grupos
- f) Definição de um projeto
  - i) Enunciado
  - ii) Prazo
  - iii) Objetivos
  - iv) Etapas
- g) Publicar anúncios
  - i) Título
  - ii) Mensagem
- h) Responder aos alunos
  - i) Mensagem
- i) Dar feedback aos alunos (projetos)
  - i) Mensagem



### 3. Alunos:

- j) Criar/juntar a grupo
  - i) Número do grupo
- k) Comunicação
  - i) Com Aluno
  - ii) Com Professor
- l) Documentos e artefactos relativos ao Projeto
- m) Períodos de disponibilidade
- n) Tarefas
  - i) Nome
  - ii) Inicio
  - iii) Fim
- o) Avaliação de elementos do Grupo

### **3.2. Dados de saída**

#### 1. Sistema:

- a) Mensagens dos alunos/docentes
- b) Anúncios do professor
- c) Trabalhos guardados
- d) Feedback do professor
- e) Dados de perfil
- f) Horários de disponibilidade dos alunos
- g) Tempo dedicado a cada tarefa

## PARTE II – PLANEAMENTO

### 4. Recursos

#### 4.1. Pessoas da equipa

Recursos Humanos:

- André Silva, 51634
- António Pinto, 52045
- Diogo Frazão, 51595
- Guilherme Nunes, 51594
- João Pinto, 50030
- Tiago Robalo, 51628
- Vasco Bento, 51636

Currículo do 2º Semestre:

- André Silva:
  - Contabilidade Geral, 6 ECTS
  - Estratégia Empresarial, 6 ECTS
  - Projeto de Tecnologias de Informação, 6 ECTS
  - Projeto de Tecnologias de Redes, 6 ECTS
  - Sistemas de Informação e Gestão, 6 ECTS
- António Pinto:
  - Contabilidade Geral, 6 ECTS
  - Estratégia Empresarial, 6 ECTS
  - Projeto de Tecnologias de Informação, 6 ECTS
  - Projeto de Tecnologias de Redes, 6 ECTS
  - Sistemas de Informação e Gestão, 6 ECTS
- Diogo Frazão:
  - Análise Exploratória de Dados Multivariados, 6 ECTS
  - Estatística, Ciência e Sociedade, 6 ECTS
  - Projeto de Tecnologias de Informação, 6 ECTS
  - Projeto de Tecnologias de Redes, 6 ECTS
  - Segurança Informática, 6 ECTS

- **Guilherme Nunes:**
  - Cartografia, 6 ECTS
  - Introdução às Probabilidades e Estatística, 6 ECTS
  - Métodos Óticos de Modelação 3D, 6 ECTS
  - Segurança Informática, 6 ECTS
  - Projeto de Tecnologias de Informação, 6 ECTS
  - Projeto de Tecnologias de Redes, 6 ECTS
  
- **João Pinto:**
  - Estatística, Ciência e Sociedade, 6 ECTS
  - Introdução às Probabilidades e Estatística, 6 ECTS
  - Laboratório de Estatística II, 6 ECTS
  - Projeto de Tecnologias de Informação, 6 ECTS
  - Projeto de Tecnologias de Redes, 6 ECTS
  
- **Tiago Robalo:**
  - Contabilidade Geral, 6 ECTS
  - Estratégia Empresarial, 6 ECTS
  - Projeto de Tecnologias de Informação, 6 ECTS
  - Projeto de Tecnologias de Redes, 6 ECTS
  - Sistemas de Informação e Gestão, 6 ECTS
  
- **Vasco Bento:**
  - Projeto de Tecnologias de Informação, 6 ECTS
  - Projeto de Tecnologias de Redes, 6 ECTS
  - Cadeira do Minor de Multimédia, 6 ECTS
  - Cadeira do Minor de Multimédia, 6 ECTS
  - Cadeira do Minor de Multimédia, 6 ECTS

#### Atividades Extracurriculares:

- André Silva: Ginásio (3 Horas – 2 vezes por semana)
- António Pinto: Desporto (3 Horas – 2 vezes por semana)
- Tiago Robalo: Trabalhador Part-Time (20 Horas Semanais)
- Vasco Bento: Ginásio (3 Horas – 2 vezes por semana)

#### Testes e Trabalhos de Disciplinas:

- Segurança Informática: Semanas 6 e 12
- Contabilidade Geral: Semana 8
- Estratégia Empresarial: Semana 14
- Sistemas de Informação e Gestão: Semanas 6, 8, 9 e 13
- Análise Exploratória de Dados Multivariados: Semana 16
- Estatística, Ciência e Sociedade: Semanas 9 e 15

Períodos de Férias:

- Carnaval: 24 a 26 de fevereiro
- Páscoa: 8 a 14 de abril
- Pausa Letiva: 30 de maio a 2 de junho
- Exames: 3 a 20 de junho

Feriados:

- Sexta-Feira Santa: 10/4/2020
- Domingo de Páscoa: 12/4/2020
- 25 de Abril: 25/4/2020
- Dia do Trabalhador: 1/5/2020
- Dia de Portugal: 10/5/2020

Para calcular a disponibilidade base de cada elemento, tivemos em consideração o número de créditos da cadeira, neste caso 6, sobre o número total de cadeiras realizadas no semestre (vezes 100 para obter a percentagem). Relativamente à disponibilidade base teremos em consideração fatores externos ao ambiente da cadeira, como atividades extracurriculares e avaliações de outras cadeiras, que irão influenciar as disponibilidades.

Datas	André	António	Diogo	Guilherme	João	Tiago	Vasco
Semana 1 (17/02 – 23/02)	30%	30%	30%	33%	40%	25%	30%
Semana 2 (24/02 – 01/03)	15%	20%	50%	25%	50%	45%	10%
Semana 3 (02/03 – 08/03)	30%	30%	30%	23%	40%	25%	30%
Semana 4 (09/03 – 15/03)	30%	30%	30%	33%	40%	25%	30%
Semana 5 (16/03 – 22/03)	30%	30%	30%	23%	40%	25%	30%
Semana 6 (23/03 - 29/03)	20%	20%	25%	20%	40%	15%	30%
Semana 7 (30/03 – 5/04)	30%	30%	30%	33%	40%	25%	30%
Semana 8 (06/04 – 12/04)	15%	15%	30%	33%	40%	10%	30%
Semana 9 (13/04 – 19/04)	30%	30%	20%	33%	40%	15%	30%
Semana 10 (20/04 – 26/04)	35%	35%	35%	30%	45%	25%	35%
Semana 11 (27/04 – 03/05)	35%	35%	35%	30%	45%	25%	35%
Semana 12 (04/05 – 10/05)	35%	35%	25%	25%	45%	25%	35%
Semana 13 (11/05 – 17/05)	20%	20%	40%	33%	40%	20%	30%
Semana 14 (18/05 – 24/05)	20%	20%	40%	33%	40%	20%	30%
Semana 15 (25/05 – 31/05)	40%	40%	40%	40%	50%	40%	40%
Semana 16 (01/06 – 07/06)	40%	40%	40%	40%	50%	40%	40%

*Tabela 1 - Tabela de Disponibilidade por Semana*

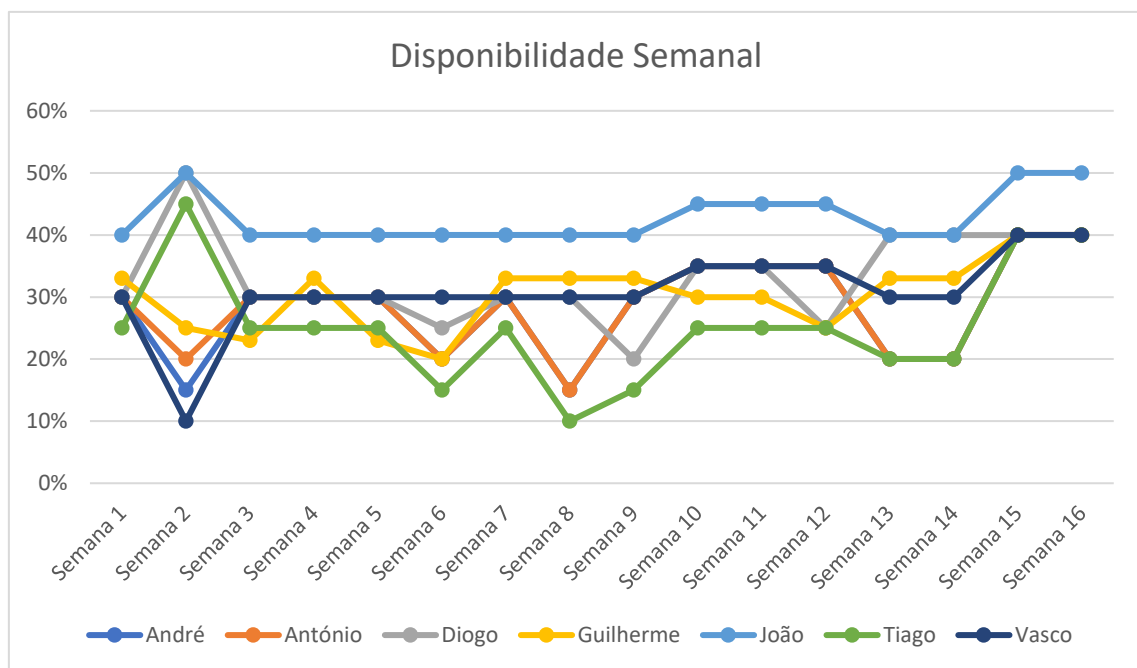


Figura 1 - Gráfico de linhas disponibilidade semanal por aluno (%)

	André	António	Diogo	Guilherme	João	Tiago	Vasco
HTML	5	5	5	5	5	5	5
CSS	4	4	4	4	3	4	4
JavaScript	3	4	5	4	4	4	4
Framework MVC	2	2	1	1	4	2	1
PHP	3	3	4	5	3	4	3
SQL	4	5	4	4	4	3	4
Segurança	3	3	3	3	3	3	3
Redes de Computadores	3	4	3	4	3	3	3
Desenho de Interfaces	4	4	4	4	4	4	5

Tabela 2 - Tabela de Competências Técnicas: 1 - Muito Fraco; 2 - Fraco; 3 - Suficiente; 4 - Bom; 5 - Muito Bom

	André	António	Diogo	Guilherme	João	Tiago	Vasco
Organização	4	4	4	4	4	4	4
Trabalho em equipa	5	4	3	3	4	4	5
Gestão de Tempo	4	3	4	4	4	3	3
Planeamento	3	4	5	4	4	4	4
Análise	3	4	4	4	4	4	4
Responsabilidade	4	5	3	4	4	4	3
Autonomia	4	4	4	5	4	4	4

*Tabela 3 - Tabela de Competências Pessoais: 1 - Muito Fraco; 2 – Fraco; 3 – Suficiente; 4 – Bom; 5 – Muito Bom*

## **4.2. Software para o sistema**

Amazon Web Services [1] – Web service que disponibiliza capacidade computacional segura e redimensionável na nuvem.

MySQL [2] – Sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface.

GitLab [3] – Sistema de controlo de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registar o histórico de edições de qualquer tipo de arquivo.

Kubernetes [4] – Plataforma open-source para gerir containers de serviços e cargas de trabalhos.

## **4.3. Ferramentas de desenvolvimento**

### **4.3.1. Hardware**

- Computadores pessoais – Pois estarão sempre disponíveis, e assim sempre se poderá trabalhar mesmo que os computadores do DI não estejam disponíveis. Além disso, os membros do grupo já se sentem confortáveis com os mesmos.
- Computadores dos laboratórios do DI e restantes – Porque estão disponíveis 24 horas por dia, mas restritos ao seu estado de ocupação.

### **4.3.2. Software**

1. Sistemas Operativos:
  - Microsoft Windows – Utilizamos este sistema operativo uma vez que é o mais utilizado, e está disponível tanto nos computadores pessoais como nos computadores dos laboratórios da Faculdade. Além disso é aquele que todos os membros do grupo mais dominam.
  - Linux (Xubuntu) - Iremos utilizar este sistema operativo, uma vez que se apresenta vantajoso para os membros do grupo utilizar o Terminal do Linux, dado que foi este que foi lecionado nas unidades curriculares.
2. Editores de Texto – Uso dependente do gosto pessoal de cada membro do grupo:
  - Atom
  - Sublime Text
  - Notepad++
  - Brackets
3. Ambiente de Desenvolvimento:



- PhpStorm [5] - Este ambiente é utilizado porque suporta o desenvolvimento em simultâneo de diversas linguagens de programação. Sendo possível a ligação direta á base de dados e é possível executar os ficheiros localmente funcionando como um servidor.
4. Web Browsers:
- Google Chrome [6] – Este browser será utilizado porque atualmente é o mais rápido, eficiente e usado do mercado. A web engine deste browser é utilizada por outros browsers (Opera) na sua versão open-source (Chromium).

#### **4.3.3. Linguagens de Programação**

Serão usadas as linguagens lecionadas e apreendidas durante a licenciatura. Sendo também as mais utilizadas e eficientes para o desenvolvimento Web:

5. HTML5 [7]
6. CSS [8]
7. JavaScript [9]
8. PHP [10]
9. SQL [11]

#### **4.3.4. Bibliotecas Externas**

1. Bootstrap [12] – Sistema de auxílio ao design e formatação web, sendo bastante eficaz.
2. Jquery [13] – Em alguns casos esta biblioteca auxilia e facilita o desenvolvimento de funções.
3. AJAX [14] – Permite alterar partes específicas de uma página web assincronamente.

#### **4.3.5. Frameworks**

1. Laravel [15] – Iremos utilizar esta framework para auxiliar a construção do projeto, usando as suas bibliotecas.

#### **4.3.6. Serviços de Comunicação**

1. WhatsApp [16] – Sistema mais utilizado para comunicação entre elementos de grupo. Além disso permite a transferência de todos os tipos de ficheiro (incluindo scripts)
2. Discord [17] – Sistema utilizado para auxiliar a comunicação e a realização do trabalho a partir de casa.

3. Skype [18] – Sistema utilizado para auxiliar a comunicação e a realização do trabalho a partir de casa.

## 5. Estimativas

### 5.1. Esforço disponível

Deve ser calculado o esforço que a equipa pode dedicar ao projeto, com base nas disponibilidades de cada pessoa indicadas na secção 4.1, e na duração do projeto, sendo esta última dependente da data prevista de início e de conclusão.

$$\text{Disponibilidade} = \frac{\text{Créditos da Cadeira} \times 100}{\text{Créditos do Semestre}}$$

Disponibilidade	
André	$(12 \times 100)/30 = 40\%$
António	$(12 \times 100)/30 = 40\%$
Diogo	$(12 \times 100)/30 = 40\%$
Guilherme	$(12 \times 100)/36 = 33.33\%$
João	$(12 \times 100)/30 = 40\%$
Tiago	$(12 \times 100)/30 = 40\%$
Vasco	$(12 \times 100)/30 = 40\%$

Tabela 4 - Tabela de Disponibilidade

$$\text{Esforço} = \text{Número de Pessoas} \times \text{Duração (PM)}$$

Disponibilidade	$(0.40 \times 6) \text{ e } (0.33 \times 1)$
Nº de Pessoas	$(0.40 \times 6) + (0.33 \times 1) = 2.73$
Duração em Meses	4
Esforço Total (PM)	$2.73 \times 4 = 10.92$

Tabela 5 - Tabela do Cálculo do Esforço Disponível

## 5.2. Dados históricos

Apresentar, para cada elemento da equipa, dados relativos a projetos passados, indicando uma breve descrição do projeto, a linguagem de programação utilizada, o total de linhas de código, a duração, o esforço, e a produtividade. Estes dados podem ser compilados em uma ou mais tabelas para facilitar o cálculo de valores agregados, como médias, mínimos e máximos.

Se existir muita repetição de dados, por exemplo, se vários membros da equipa tiverem pertencido aos mesmos grupos que realizaram os projetos passados, a organização deste capítulo deve passar a ser feita de forma cronológica ou segundo outro critério, como o tamanho do projeto.

No final desta secção devem ser apresentados histogramas das linhas de código e das produtividades de todos os projetos passados considerados.

### Descrição de cada projeto

**Programação I** – Construção de software que apoia o direcionamento de pedidos de ajuda usado pela empresa NSHF para gerir a atribuição de pedidos de ajuda por chat aos operadores do seu centro de contacto.

Linguagens de Programação Utilizadas: Python.

Número de Elementos por grupo: 2.

**Introdução as Tecnologias Web** – O objetivo deste projeto é desenvolver o jogo da batalha naval e o seu site, usando as tecnologias lecionadas na disciplina

Linguagens de Programação Utilizadas: HTML, CSS e JavaScript.

Número de Elementos por grupo: 3.

**Programação II** – Neste projeto vai implementar o software clustering Aliens. É um software que permite obter o agrupamento de objetos de acordo com as suas características apresentadas num vetor.

Linguagens de Programação Utilizadas: Python.

Número de Elementos por grupo: 2.

**Base de Dados** – Modelação do funcionamento de uma plataforma de concertos, incluindo criação de uma base de dados.

Linguagens de Programação Utilizadas: SQL.

Número de Elementos por grupo: 4.

**Interação com Computadores** – Realização de um protótipo para uma interface da aplicação SmartHome.

Linguagens de Programação Utilizadas: HTML, CSS e JavaScript.

Número de Elementos por grupo: 3.

**Programação Centrada em Objetos** – Realizar um esboço de uma aplicação de ajuda à partilha de despesas num grupo de utilizadores. Cada utilizador registado pode criar um “split” i.e. uma atividade à qual serão associadas despesas, adicionar despesas com lista dos beneficiários da desta, consultar balanço associado, ver o saldo de cada utilizador.

Linguagens de Programação Utilizadas: Java.

Número de Elementos por grupo: 2.

**Sistemas Operativos** – Realização de um trabalho envolvendo a criação de processos/threads e a comunicação entre processos/threads envolvendo também a manipulação de ficheiros e o tratamento de sinais, tempo e alarmes.

Linguagens de Programação Utilizadas: Python.

Número de Elementos por grupo: 2.

**Análise e Desenho de Software** – Pretende-se que contribua para o desenvolvimento iterativo de um sistema de gestão de Leilões Online.

Linguagens de Programação Utilizadas: Java.

Número de Elementos por grupo: 3.

**Aplicações Distribuídas** – O objetivo geral do projeto será concretizar um gestor de pedidos simultâneos a recursos e processamento destes em exclusão mútua.

Linguagens de Programação Utilizadas: Python e SQL.

Número de Elementos por grupo: 3.

**Aplicações e Serviços de Web** – Realizar uma aplicação web para coleção e troca de cartas. O propósito final da aplicação deve ser um site que permita que vários utilizadores registados possam iniciar coleções diversas de cartas.

Linguagens de Programação Utilizadas: HTML, CSS, JavaScript e PHP.

Número de Elementos por grupo: 3.

**Sistemas Inteligentes** – Modelação do problema do pacman, formulação do paradigma do espaço de estados, recolha de da dose comparação de algoritmos.

Linguagens de Programação Utilizadas: Python.

Número de Elementos por grupo: 3.

A tabela seguinte representa as linhas de código dos projetos de cada elemento do grupo, consoante a disciplina. Alguns valores são iguais pois os elementos pertenciam aos mesmos grupos, porém, como os grupos não foram uniformes, havendo constituições diferentes para certas cadeiras, por forma a facilitar a apresentação dos dados, a tabela apresenta os alunos por projeto da cadeira. No final da tabela está presente a média de todos os elementos do grupo, por cadeira.

	PROG1	ITW	PROG2	BD	IC	PCO	SO	ADS	AD	ASW	SI
<b>André</b>	224	1021	276	349	17595	774	300	617	1072	3600	933
<b>António</b>	217	1568	220	288	17595	847	268	617	1072	3600	933
<b>Diogo</b>	195	867	243	349	10719	808	268	856	1011	3222	821
<b>Guilherme</b>	217	1568	276	288	17595	847	268	617	1072	3600	933
<b>João</b>	-	3924	-	-	-	-	-	3000	1131	15000	-
<b>Tiago</b>	220	1568	295	288	10719	774	300	856	1011	3222	821
<b>Vasco</b>	215	2349	220	288	10719	808	300	856	1011	3222	821
<b>Média</b>	215	1838	255	308	14157	810	284	1060	1054	5067	887

*Tabela 6 – Tabela com Linhas de código por projeto em que cada membro participou*

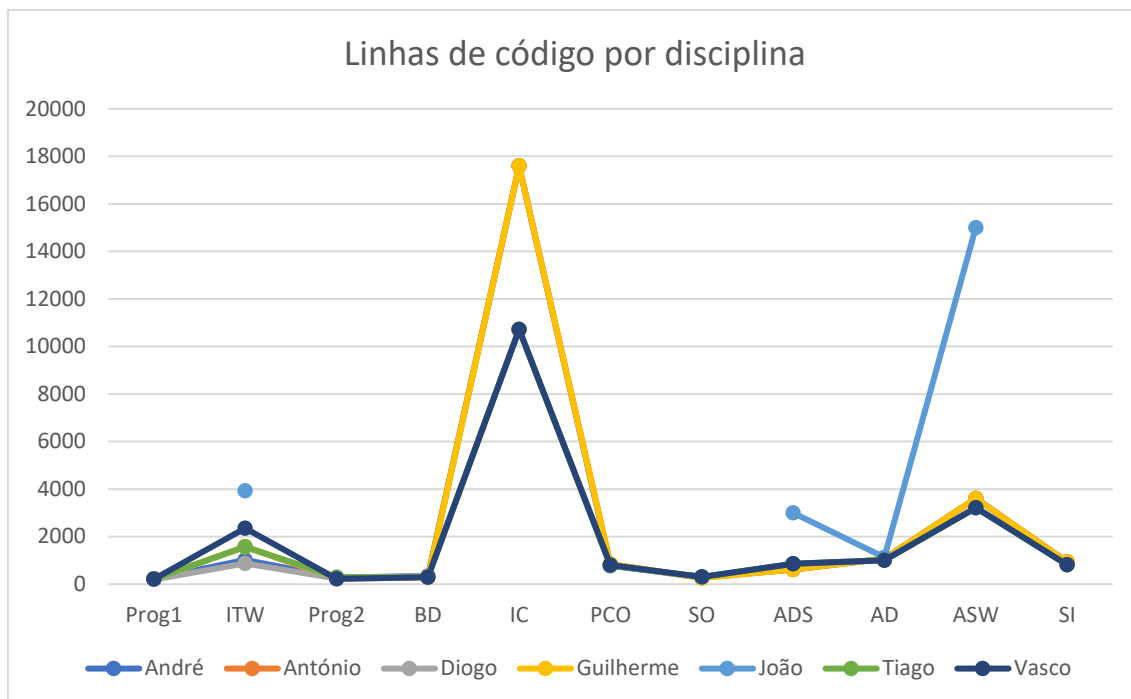


Figura 2 - Gráfico de linhas com linhas de código por projeto em que cada membro participou

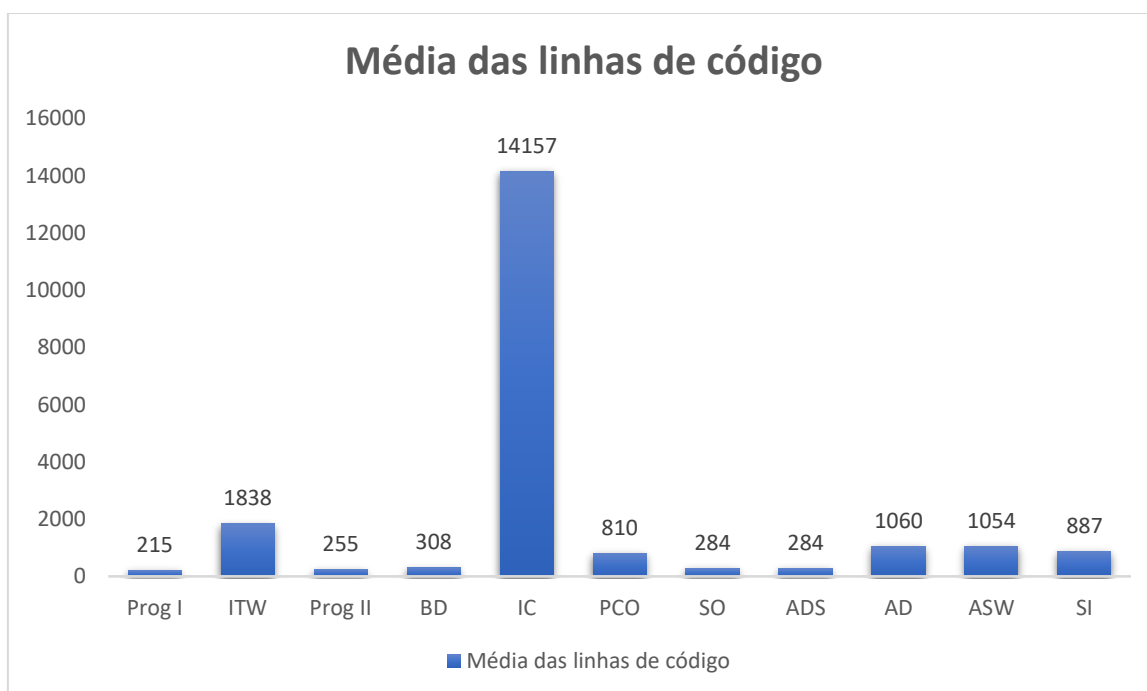


Figura 3 - Gráfico de barras com média de linhas de código por unidade curricular

A tabela abaixo apresenta uma média, de todos os elementos, das linhas de código relacionando a cadeira com a linguagem. No final foi calculada o número total das médias de linhas de código de todas as cadeiras, podendo assim ter uma noção mais aprofundada sobre a quantidade de linhas de código, de cada linguagem, até então codificadas, percebendo as linguagens com as quais os intervenientes do grupo estão mais familiarizados.

	Python	Java	SQL	HTML	CSS	JavaScript	PHP
PROG1	215	-	-	-	-	-	-
ITW	-	-	-	494	564	637	-
PROG2	255	-	-	-	-	-	-
BD	-	-	308	-	-	-	-
IC	-	-	-	8880	2280	2997	-
PCO	-	810	-	-	-	-	-
SO	284	-	-	-	-	-	-
ADS	-	1060	-	-	-	-	-
AD	996	-	58	-	-	-	-
ASW	-	-	-	-	703	687	3533
SI	877	-	-	-	-	-	-
Total	2627	1870	366	9374	3547	4321	3533

*Tabela 7 – Tabela de Linhas de código por projeto por linguagem*



De seguida é apresentado o esforço (Pessoas \* Prazo) e a produtividade (Linhas de código / Esforço) de todas as cadeiras, por linguagem e no total. No cálculo do esforço, foi multiplicado 0.2 pelo número de pessoas, pois foi a disponibilidade dos intervenientes para todos os projetos, tendo em conta o número de cadeiras por semestre.

Na seguinte tabela, o prazo tido em conta foi a duração total do projeto, isto é, desde o dia em que o enunciado foi publicado até ao dia da entrega do projeto. Foi tomada esta abordagem pois o não temos dados que permitam obter o tempo concreto de duração de cada projeto, ou seja desde o início da sua execução até ao seu término.

Assim sendo, a produtividade obtida por cada unidade curricular será inferior à que seria obtida se tivéssemos em conta a duração real da execução dos projetos.

	Linguagem	Prazo (Mês)	LOC	Elementos	Esforço (PM)	Produtividade (LOC/Esforço)
PROG1	Python	1.5	215	2	$(2*0.2)*1.5 = 0.6$	358
ITW	HTML	3	494	3	$(3*0.2)*3 = 1.8$	274
	CSS		564			313
	JavaScript		637			354
	Total		1838			1021
PROG2	Python	2	255	2	$(2*0.2)*2 = 0.8$	319
BD	SQL	0.5	308	4	$(4*0.2)*0.5 = 0.4$	770
IC	HTML	2	8880	3	$(3*0.2)*2 = 1.2$	7400
	CSS		2280			1900
	JavaScript		2997			2498
	Total		14157			11798
PCO	Java	2	810	2	$(2*0.2)*2 = 0.8$	1013
SO	Python	2	284	3	$(3*0.2)*2 = 1.2$	237
ADS	Java	1	1060	3	$(3*0.2)*1 = 0.6$	1767
AD	Python	3	996	3	$(3*0.2)*3 = 1.8$	553

	SQL		58			32
	Total		1054			586
ASW	PHP	3	3533	3	$(3*0.2)*3 = 1.8$	1963
	CSS		703			391
	JavaScript		687			387
	Total		5067			2815
SI	Python	1	887	3	$(3*0.2)*1 = 0.6$	1478

Tabela 8 – Tabela com as linhas de código, duração, esforço e produtividade de cada projeto

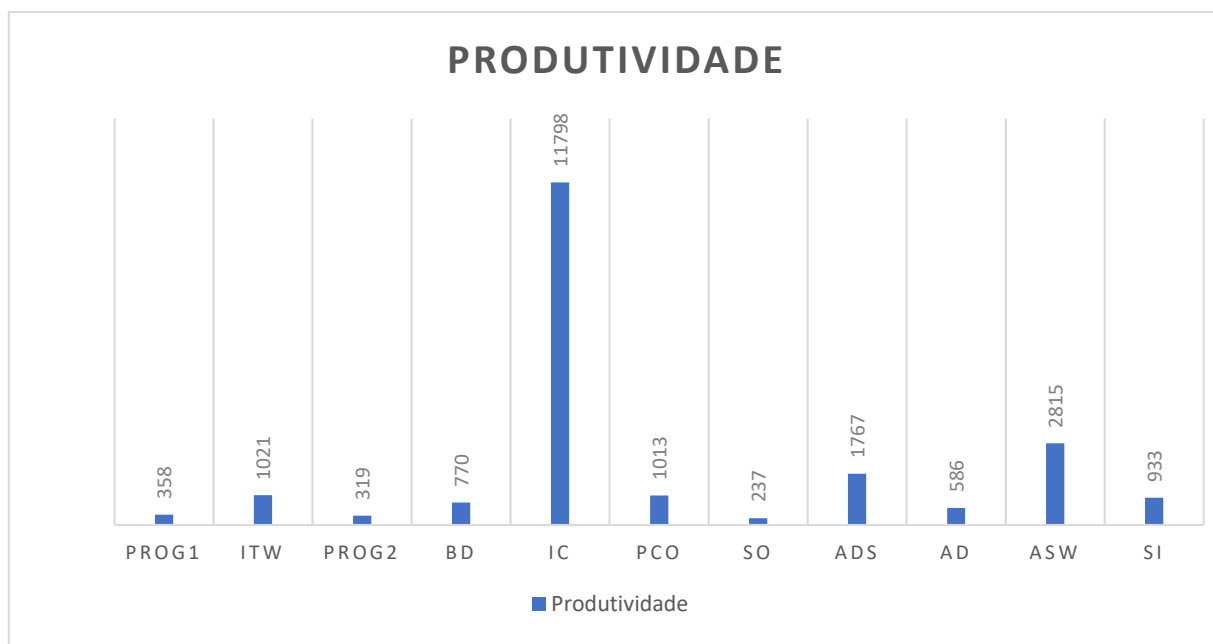


Figura 4 - Gráfico de barras com a produtividade total por unidade curricular

Por fim, podemos calcular a Produtividade final, sendo que esta é a soma das produtividades totais de todas as unidades curriculares.

**Produtividade Final = 2015 LOC/PM**

### 5.3. Estimativas baseadas em linhas de código

Nesta secção decompomos o produto em diversas componentes para aumentar a probabilidade de calcular estimativas próximas do valor real. Cada componente é ainda decomposta em atividades mais simples (funções). As linhas de código destas atividades podem ser estimadas a partir de outras funções semelhantes que já tenhamos desenvolvido em projetos passados.

#### 5.3.1. Requisitos Funcionais

##### Alunos

- RF-1    Efetuar login de utilizador
  - Formulário com campos para o aluno
  - Verificação dos dados do aluno
  - Autenticação do aluno
  - Conceder acesso à aplicação
  
- RF-2    Efetuar logout de utilizador
  - Limpar os dados de sessão do utilizador
  
- RF-3    Visualizar perfil de utilizador
  - Verificação das permissões de acesso à página
  - Obtenção e apresentação da informação do aluno
  
- RF-4    Editar perfil do utilizador
  - Autenticação do aluno
  - Verificação dos novos dados
  - Atualização dos dados na base de dados
  
- RF-5    Criar grupo de trabalho para a unidade curricular
  - Verificação do estado do grupo
  - Conceder acesso ao grupo
  - Registar os novos dados na base dados
  
- RF-6    Sair de um grupo de trabalho de uma unidade curricular
  - Verificação se a saída é possível na presente data
  - Atualização dos dados na base de dados
  
- RF-7    Pedir para juntar a grupo existente
  - Envio de mensagem para os elementos do grupo

- Atualização da base de dados caso este seja aceite
- RF-8 Gerir pedidos para juntar a grupo
- Responder aos pedidos feitos pelos elementos que pretendem integrar o grupo
- RF-9 Comunicar entre elementos do grupo
- Envio e receção de mensagens referentes a um grupo
  - Guardar o histórico de mensagens
- RF-10 Comunicar com outros alunos
- Envio e receção de mensagens a outros alunos
  - Guardar o histórico de mensagens
- RF-11 Comunicar com o professor
- Envio e receção de mensagens
  - Guardar o histórico de mensagens
- RF-12 Gerir documentos na área de trabalho do grupo
- Inserção de um novo ficheiro
  - Remoção de um ficheiro da área de trabalho
  - Associação de uma submissão a um aluno assim como a data
- RF-13 Submeter versão final dos trabalhos
- Envio dos ficheiros para uma área de projetos finalizados
- RF-14 Criar tarefas referentes a um trabalho
- Inserção e remoção de tarefas numa tabela
  - Adicionar um registo com a data de início da tarefa
- RF-15 Terminar tarefas referentes a um trabalho
- Calcular o tempo de duração da tarefa
- RF-16 Gerir tarefas referentes a um trabalho
- Alterar estado da tarefa
  - Mudar propriedades da tarefa
- RF-17 Registrar horário de disponibilidade
- Indicação do horário de disponibilidade num calendário semanal
  - Alteração do horário de disponibilidade

- Conjugação de todas as disponibilidades dos elementos dos grupos
- RF-18 Ver horário de disponibilidade dos elementos do grupo
- Apresentação do horário das disponibilidades de todos os integrantes do grupo
- RF-19 Atribuir pontuações/avaliação ao trabalho dos restantes elementos do grupo
- Atribuição de pontuação aos restantes elementos do grupo
  - Inserção das avaliações na base de dados

### **Professores**

- RF-20 Efetuar login de docente
- Formulário com campos para o professor
  - Verificação dos dados do professor
  - Autenticação do professor
  - Conceder acesso à aplicação
- RF-21 Efetuar logout de docente
- Limpar os dados de sessão do utilizador
- RF-22 Definir regras para constituição de grupos
- Número de elementos por grupo
  - Condição de pertencerem à mesma turma
  - Definição dos prazos de constituição e alteração de grupos
- RF-23 Definir número de grupos possíveis
- Indicação do número de slots de grupos possível
- RF-24 Publicar anúncios com informações referentes ao trabalho
- Escrita na base de dados dos anúncios
  - Permitir que os alunos possam visualizar estes anúncios
- RF-25 Estipular etapas para os projetos (enunciado do projeto, objetivos, entregáveis e prazos)
- Disponibilizar os enunciados dos trabalhos
  - Informar quais os objetivos e entregáveis do trabalho
  - Criar áreas para submissão de etapas com prazos

- RF-26 Consultar os documentos e outros artefactos que vão sendo produzidos por cada grupo em cada projeto
- Permitir a visualização das submissões dos alunos
- RF-27 Responder a dúvidas dos grupos
- Visualizar as questões dos alunos
  - Responder às questões guardando as respostas
- RF-28 Comunicar com os alunos
- Envio e receção de mensagens
  - Guardar o histórico de mensagens
- RF-29 Comunicar com os outros professores
- Envio e receção de mensagens
  - Guardar o histórico de mensagens
- RF-30 Dar feedback a cada grupo em cada etapa
- Criar um campo com o feedback do docente, permitindo que o aluno o visualize

### **Administradores do Sistema**

- RF-31 Assegurar a existência de dados dos professores
- Criação de um modelo de dados
  - Preenchimento da tabela com os dados relativos aos professores
  - Desenho conceptual e lógico
- RF-32 Assegurar a existência de dados dos alunos
- Criação de um modelo de dados
  - Preenchimento da tabela com os dados relativos aos alunos
  - Desenho conceptual e lógico
- RF-33 Criação de novas unidades curriculares
- Possibilidade de adicionar uma unidade curricular nova caso esta seja introduzida num curso
- RF-34 Assegurar a existência de dados das unidades curriculares
- Criação de um modelo de dados
  - Preenchimento da tabela com os dados relativos às unidades curriculares
  - Desenho conceptual e lógico

- RF-35 Criação de novas turmas
- Possibilidade de criar uma turma
- RF-36 Assegurar a existência de dados das turmas
- Criação de um modelo de dados
  - Preenchimento da tabela com os dados relativos às turmas
  - Desenho conceptual e lógico
- RF-37 Importação de dados em ficheiros ou outras fontes
- O sistema de gestão académica aceita importação de ficheiros de outros tipos de dados, nomeadamente CSV, XML. SQL
- RF-38 Assumir permissões distintas com base no perfil de utilizador
- Apenas o corpo docente da correspondente unidade curricular pode criar/administrar trabalhos
  - Alunos só tem acesso aos trabalhos das cadeiras em que estão inscritos.
  - Alunos só tem acesso aos seus próprios dados (mensagens, ficheiros) e dos grupos e trabalhos a que pertencem
- RF-39 Retirar permissões
- Restringir as permissões de uma pessoa

### 5.3.2. Requisitos não funcionais

#### Persistência e disponibilidade

- a. Uso de um Sistema de Base de Dados, baseado em MySQL
- b. Atualização de software rápida, por forma a minimizar o período de manutenção

#### Segurança

- a. Uso de Credenciais de segurança
- b. Uso de Criptografia
- c. Acesso ao servidor através de login
- d. Proteção contra acesso ilícito de dados

#### Baixo custo e Escalabilidade

- a. A capacidade da cloud escolhida varia automaticamente por forma a interferir com a escalabilidade

#### Usabilidade

- a. Interface simples e intuitiva, feita com base em testes de aceitação
- b. Esconder a complexidade do programa aos olhos do utilizador

#### Suporte

- a. Testes com variados tipos de dispositivos para garantir o seu bom funcionamento

#### Fiabilidade

- a. Várias cópias da base de dados
- b. Atualização das cópias num período fixo de acordo com o calendário
- c. Manutenção do hardware recorrente

$$Estimativa\ Final = \frac{E_{otimista} + 4 \times E_{provável} + E_{pessimista}}{6}$$



	Otimista	Provável	Pessimista	Final
RF1	130	150	200	155
RF2	50	60	100	65
RF3	100	150	170	145
RF4	100	110	130	82
RF5	170	200	230	200
RF6	100	140	170	138
RF7	90	150	190	147
RF8	120	150	180	150
RF9	300	400	500	287
RF10	300	350	400	350
RF11	300	350	400	350
RF12	180	220	300	227
RF13	100	120	150	122
RF14	90	100	130	103
RF15	30	40	60	42
RF16	90	110	120	108
RF17	400	500	700	517
RF18	300	350	450	358
RF19	70	80	100	82
RF20	130	150	200	155
RF21	20	40	60	40
RF22	100	140	200	143

RF23	70	100	150	103
RF24	120	150	200	153
RF25	110	130	150	130
RF26	120	150	250	162
RF27	70	100	150	103
RF28	300	350	400	350
RF29	300	350	400	350
RF30	50	70	100	72
RF31	120	140	200	147
RF32	120	140	200	147
RF33	40	50	70	52
RF34	120	140	200	147
RF35	50	80	90	77
RF36	120	140	200	147
RF37	300	400	600	417
RF38	150	170	220	175
RF39	90	110	120	108
Segurança	300	400	600	417

Tabela 9 - Estimativa final das componentes relativas aos requisitos

**Total LOC = 7223**

$$Esforço = \frac{LOC}{Produtividade} = \frac{7223}{2015} \approx 3.6 PM$$

O valor 2015 corresponde à produtividade final, calculada na secção anterior.

#### 5.4. Estimativas baseadas no modelo COCOMO E<sub>otimista</sub>

Através do valor total de LOC obtido na secção anterior, vamos estimar, utilizando o Modelo Intermédio do COCOMO, o esforço e duração necessários para o desenvolvimento do projeto em função da dimensão do software (LOC), visto que o grupo estimou 6419 e este valor encontra-se abaixo das 50 mil linhas de código, o tipo escolhido vai ser o orgânico, e de um conjunto de fatores subjetivos que irão integrar um fator (EAF) que multiplicará pelo esforço, para que este tenha em conta esses fatores. Os valores dos fatores estão sublinhados na tabela abaixo. [19, pp. 744-753]

$$EAF = Me1 \times Me2 \times \dots \times Me15$$

$$\text{Esforço} = a \times KLOC^b \times EAF$$

$$\text{Duração} = c \times \text{Esforço}^d$$

Projecto	a	b	c	d
Orgânico	2.4	1.05	2.5	0.38
Semi-ind.	3	1.12	2.5	0.35
Embebido	3.6	1.2	2.5	0.32

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
<b>Product attributes</b>						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
<b>Hardware attributes</b>						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
<b>Personnel attributes</b>						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
<b>Project attributes</b>						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

$$EAF = 1^9 \times 1.8 \times 1.15 \times 1.06 \times 1.19 \times 0.86 = 1.35$$

$$\text{Esforço} = 2.4 \times 7.223^{1.05} \times 1.35 \cong 26\text{PM}$$

$$\text{Duração} = 2.5 \times 25.83^{0.38} \cong 9 \text{ Meses}$$

$$\text{Produtividade} = \frac{7223}{26} \cong 278 \text{ LOC/PM}$$

$$\text{Nº Elementos} = \frac{26}{9} \cong 3 \text{ Pessoas}$$

## 5.5. Análise Crítica

Na secção 5.1 foi estimada a disponibilidade, com base na relação dos créditos das cadeiras referentes ao projeto e as do semestre, para possibilitar o cálculo do esforço total, foi também atribuída uma duração de 4 meses. Consoante a disponibilidade dos elementos do grupo, foi obtido um valor de 2.7 pessoas, o que dividindo pela duração do projeto, resulta num valor de esforço total de 11 PM. Na secção 5.3 foi calculado o esforço com base na decomposição. Após a decomposição do projeto em componentes, foi estimado que seriam necessárias 7223 linhas de código. Sabendo as linhas de código apenas faltava a disponibilidade, que tinha sido calculada na secção anterior. No 5.2, foi calculada a produtividade com base em registos históricos, sendo que após a recolha de toda a informação necessária, referente a cada aluno, e feitas as médias para que o valor da produção fosse transversal, abrangendo todos os elementos do grupo e todas as cadeiras até então realizadas, foi obtido um valor de produtividade final igual a 2015. Posto isto já são conhecidos todos os atributos necessários para o cálculo do esforço no 5.3, que dividindo o número de linhas de código estimadas (7223) pela produtividade (2015) resulta o esforço de 3,6 PM.

Podemos afirmar que entre a secção 5.1 e 5.3 existe uma grande discrepância, o que se deve ao facto de a primeira apenas ter em conta a duração e a disponibilidade relativa (apenas tem em consideração as cadeiras) sem ter qualquer suporte de dados históricos, que são um fator importante no cálculo do esforço, daí podermos concluir que à partida o valor estimado em 5.3 será mais fiável, apesar da complexidade requerida nos projetos anteriores ser menor assim como o conhecimento da equipa face à abordagem dos projetos, o que se pode tornar num fator relevante, influenciando algum desvio nos valores estimados.

Através do modelo COCOMO, foi obtido um esforço de 26 PM, o que ainda assim é bastante superior ao valor obtido na secção 5.1, sendo que este resultado pode ser suportado pelo facto de o COCOMO considerar uma equipa semelhante à primeira estimação (3 e 2.7) e o esforço ser mais do dobro (10.92 PM na secção 5.1 e 26 PM na 5.4). A duração do projeto estimada pelo modelo COCOMO foi excedida por 5 meses (9 meses no total), valor que pode ser justificado pelo facto de as linhas de código estimadas serem 7223 e o tipo abordado no COCOMO (orgânico) ser para projetos que vão desde as 2 mil linhas de código até às 50 mil, posto isto podemos concluir que pelo facto de o COCOMO estimar com um tamanho de equipa mais pequeno (3) do que a realidade (7), influencia os valores da produtividade e do esforço. Outro fator que pode ter a ver com a discrepância de valores em relação ao 5.3 é o facto do valor da produtividade, estimado no modelo COCOMO, estar deflacionado dado à diversidade da natureza dos projetos.

## 6. Processo de desenvolvimento de *software*

O processo de desenvolvimento de software que escolhemos para desenvolver o projeto de PTI e PTR é o **Scrum**. Este processo é composto por um backlog, que se trata de uma lista de requisitos do cliente, ordenada por prioridade. O backlog é avaliado e ordenado pelo Product Owner que iremos definir.

O product backlog apresenta subconjuntos chamados de sprint backlogs, que é uma unidade de trabalho necessária para completar um item do backlog.

Desta forma, em cada Sprint é designada uma equipa que irá trabalhar em itens prioritários do backlog, num prazo fixo definido anteriormente, como por exemplo entre os 15 e 30 dias. O Sprint backlog não poderá ser alterado durante a execução do Sprint.

Durante cada Sprint irão ser realizadas curtas reuniões diárias com todos os membros de grupo possíveis, e irá ser feito um ponto de situação do que foi feito desde a última reunião, se foram encontradas dificuldades e o que cada membro irá desenvolver até à próxima reunião. O processo irá continuar a sua iteração até estarem terminados todos itens do backlog.

No fim de cada Sprint será apresentado uma demo, um incremento, ao cliente para demonstração e avaliação. A demo não tem de conter todas as funcionalidades planeadas para o produto, mas sim as previstas no seu Sprint.

Este modelo adequa-se ao projeto que vamos realizar no 2º semestre nas unidades curriculares de PTI e PTR porque apresenta algumas vantagens, tais como:

1. Este processo permite aumentar o ritmo de trabalho e eliminar o desperdício de tempo pois com o a existência de um backlog e respetivos sprints será possível que todos os elementos do grupo vão desenvolvendo o projeto.
2. Além disso, as reuniões permitem manter o grupo atualizado acerca do estado do projeto, o que consequentemente irá melhorar a comunicação e a motivação do grupo.
3. Será possível visualizar o progresso do projeto, de uma forma mais clara e transparente, devido à divisão por Sprints.
4. Este processo iterativo permite que a equipa vá melhorando ao longo do tempo.
5. Com a divisão em Sprints, o desenvolvimento é feito de forma fragmentada e subdividida, o que permitirá que seja mais fácil detetar erros e assim corrigi-los da forma mais rápida possível. Isto irá reduzir os riscos associados ao desenvolvimento do projeto.
6. Permite que seja apresentada mais valor ao cliente e desta forma, este irá acompanhar o projeto de uma forma mais próxima, pois em cada Sprint teremos uma versão nova do projeto.

7. Com um maior acompanhamento por parte do cliente, o projeto irá apresentar mais qualidade pois irá estar mais próximo dos requisitos do cliente.

Desvantagens:

1. Uma das desvantagens é consequência de uma vantagem, se a comunicação falhar é possível que o projeto apresente atrasos e menos qualidade, pois pode haver membros que estejam desenquadrados com o estado atual do projeto.
2. O facto de o projeto estar dividido numa tentativa de ser ágil, pode levar a equipa a perder a perspetiva do projeto como um todo, o que pode levar a falhas na hora de encaixar todos os Sprints do projeto.

A nossa implementação do modelo de processo irá decorrer num período previsto de 14 sprints, equivalentes ao número de Work Packages. Cada Sprint irá decorrer num intervalo entre 3 a 30 dias.

De acordo com o processo de desenvolvimento de software atribuímos assim as seguintes funções:

Stake Holders/Business Owner: docentes da unidade curricular;

Scrum Team:

Product Owner – António Pinto;

Scrum Master – Guilherme Nunes;

Development Team – António Pinto, André Silva, Diogo Frazão, Guilherme Nunes, João Pinto, Tiago Robalo e Vasco Bento.

## 7. Organização da equipa

Para o desenvolvimento deste projeto iremos adotar o paradigma de organização de equipas aberto, dado que se trata de problemas complexos, em que é necessário controlo e inovação. Visto que temos pouca experiência na resolução dos mesmos, é necessária muita comunicação entre os membros da equipa e decisões baseadas em consenso.

Vamos então dividir o grupo nas seguintes equipas:

Membros da equipa de PTI:

Front-end: André Silva, Vasco Bento

Back-end: Guilherme Nunes, João Pinto

Membros da equipa de PTR – António Pinto

Membros móveis – Diogo Frazão, Tiago Robalo

Sendo que quando as tarefas da equipa de front-end de PTI terminarem, os elementos passam também a ser membros móveis, pois verificámos que existia uma grande discrepância no número de horas entre estes elementos e os restantes. Isto porque as tarefas de front-end são menos e de menor duração do que as tarefas dos restantes elementos.

Responsabilidades:

- **R** – Who is Responsible.
- **A** – Who is Accountable.
- **C** – Who is Consulted.
- **I** – Who is Informed.



Tarefas	Stake Holders / Business Owners	Product Owner	Scrum Master	Development Team
Iniciar Projeto	C	AR	I	
Planeamento / Agendamento	I	R	I	I
Gestão de Riscos	I	R	AR	CI
Gestão de Qualidade	I	R	AR	CI
Desenvolver Ferramentas de Software	I	R	I	R
Testes de Software	I	R	I	R
Scrum meetings	I	R	R	I
Manter o product backlog	I	R	I	I
Manter o sprint backlog	I	R	I	I
Demo	I	AR	I	C
Entrega do Software	C	AR	CI	CR

Tabela 10 – Matriz RACI de responsabilidades

## 8. Planeamento do projeto

### 8.1. Work packages

#### WP - 1. Desenho do sistema

Descrição: Serão realizados protótipos de baixa-fidelidade de toda a aplicação para nos guiar no desenvolvimento front-end e back-end;

Realizado por: Todos os elementos do grupo;

Entregável: Esboços finais da aplicação e modelo Entidade-Associação;

Duração: 7 dias;

Precedências: Nenhuma.

Tarefa 1. Desenho de protótipos de baixa-fidelidade de todas as interfaces da aplicação.

Tarefa 2. Storyboards.

Tarefa 3. Esquema da arquitetura da aplicação (Modelo de domínio).

Tarefa 4. Realização do desenho conceptual e lógico de base de dados (Modelo Entidade-Associação).

#### WP - 2. Implementação da base de dados

Descrição: Implementar em SQL, a base de dados previamente esboçada.

Recursos: Equipa back-end de PTI;

Entregável: Tabelas da base de dados;

Duração: 5 dias;

Precedências: WP-1.

Tarefa 1. Concretização de comandos SQL para criar as tabelas da base de dados.

#### WP - 3. Desenvolvimento da interação com o servidor

Descrição: Implementar em as interações com o servidor de forma a ligar a plataforma web ao servidor;

Recursos: Equipa de PTR + membros móveis;

Entregável: Nova versão do software onde foi realizado todo o tratamento e processamento dos dados relacionados com a interação com o servidor;

Duração: 10 dias;

Precedências: Nenhuma.

Tarefa 1. Integrar a framework Laravel.

#### WP - 4. Desenvolvimento das interfaces

Descrição: Implementar o esqueleto e o design da interface de todas as páginas da aplicação;

Recursos: Equipa front-end de PTI + membros móveis;

Entregável: Esqueleto das páginas da aplicação;

Duração: 15 dias;

Precedências: WP-1.

Tarefa 1. Criação das páginas dos utilizadores (sem login feito).

Tarefa 2. Criação das páginas dos alunos.

Tarefa 3. Criação das páginas dos docentes.

Tarefa 4. Criação das páginas dos administradores de sistema .

#### WP - 5. Funcionalidades dos administradores de sistema

Descrição: Realização de todas as tarefas relacionadas com o registo de dados no sistema por parte dos administradores de Sistema;

Realizado por: Equipa de back-end PTI + membros móveis;

Entregável: Tabelas da base de dados preenchidas com os dados;

Duração: 20 dias;

Precedência: WP-1, WP-2, WP-3, WP-4, WP-5.

Tarefa 1. Realizar as funcionalidades para inserir os dados dos alunos, docentes e das unidades curriculares nas tabelas.

Tarefa 2. Realizar a funcionalidade para importar dados através de ficheiros.

Tarefa 3. Realizar as funcionalidades para atribuição/remoção de permissões aos utilizadores.

#### WP - 6. Login

Descrição: Implementar as operações de verificação das credenciais dos utilizadores;

Recursos: Equipa back-end de PTI + membros móveis;

Entregável: Possibilidade de efetuar login através da aplicação;

Duração: 3 dias;

Precedências: WP-1, WP-2, WP-3.

Tarefa 1. Realizar as verificações na base de dados das credenciais inseridas pelo utilizador.

#### WP - 7. Funcionalidades dos alunos

Descrição: Realização de todas as tarefas relacionadas com as funcionalidades que os alunos poderão realizar ao utilizar a nossa aplicação;

Realizado por: Equipa de back-end PTI + membros móveis;

Entregável: Base de dados preenchida com os horários de disponibilidade dos alunos, com as avaliações feitas pelos membros do grupo;

Duração: 30 dias;

Precedência: WP-1, WP-2, WP-3, WP-4, WP-5, WP-6.

Tarefa 1. Realização das funcionalidades para criar/juntar ou sair de um grupo.

Tarefa 2. Realização das funcionalidades para pedir para juntar a um grupo e gerir os pedidos.

Tarefa 3. Realização das funcionalidades para inserir documentos e artefactos.

Tarefa 4. Realização das funcionalidades para consultar a disponibilidade dos outros elementos do grupo.

Tarefa 5. Realização das funcionalidades para inserir a disponibilidade do aluno.

Tarefa 6. Realização das funcionalidades para atribuir pontuações/avaliação ao trabalho dos restantes elementos.

#### WP - 8. Funcionalidades dos docentes

Descrição: Realização de todas as tarefas relacionadas com as funcionalidades que os docentes poderão realizar ao utilizar a nossa aplicação;

Recursos: Equipa back-end de PTI + membros móveis;

Entregável: Base de dados preenchida com as regras de constituição de grupos e número de grupos possíveis, com os anúncios e os trabalhos submetidos pelos alunos;

Duração: 20 dias;

Precedências: WP-1, WP-2, WP-3, WP-4, WP-5, WP-6, WP-7.

Tarefa 1. Realização das funcionalidades para definir as regras de constituição de grupos e número de grupos possíveis.

Tarefa 2. Realização das funcionalidades para publicar anúncios.

Tarefa 3. Realização das funcionalidades para consultar os documentos e artefactos submetidos pelos alunos.

Tarefa 4. Realização das funcionalidades para dar feedback a cada grupo sobre os trabalhos submetidos.

#### WP - 9. Comunicação

Descrição: Realização de todas as tarefas relacionadas com a comunicação entre utilizadores;

Realizado por: Equipa back-end de PTI + membros móveis;

Entregável: Um chat de troca de mensagens entre utilizadores;

Duração: 3 dias;

Precedência: WP-1, WP-2, WP-3, WP-4, WP-5, WP-6, WP-7, WP-8.

Tarefa 1. Realização das funcionalidades de troca de mensagens entre utilizadores.

WP - 10. Implementação de segurança

Descrição: Garantir a segurança do Sistema;

Realizado por: Equipa de PTR;

Entregável: Sistema robusto e seguro, garantindo privacidade e confidencialidade de dados;

Duração: 15 dias;

Precedências: WP-2, WP-3, WP-4, WP-5, WP-6, WP-7, WP-8, WP-9.

WP - 11. Escalabilidade

Descrição: Sistema escalável;

Realizado por: Equipa de PTR + Membros móveis;

Entregável: Garantir que o sistema está pronto para alta replicação;

Duração: 15 dias;

Precedências: WP-2, WP-3, WP-4, WP-5, WP-6, WP-7, WP-8, WP-9.

WP - 12. Disponibilidade

Descrição: Garantir disponibilidade do sistema mesmo com alta afluência;

Realizado por: Equipa de PTR + Membros móveis;

Entregável: Sistema sem falhar quando altamente carregado;

Duração: 15 dias;

Precedências: WP- 2, WP-3, WP-4, WP-5, WP-6, WP-7, WP-8, WP-9.

WP - 13. Usabilidade

Descrição: Garantir as funções de usabilidade do sistema;

Recursos: Toda a equipa;

Entregável: Heurísticas e funções de usabilidade garantidas na interface;

Duração: 15 dias;

Precedências: WP-2, WP-3, WP-4, WP-5, WP-6, WP-7, WP-8, WP-9.

WP - 14. Testes

Descrição: Realização de testes de modo a determinar se o sistema está pronto;

Recursos: Toda a equipa;

Duração: 3 dias;

Precedências: WP-3, WP-4, WP-5, WP-6, WP-7, WP-8, WP-9, WP-10, WP-11, WP-12, WP-13.

Tarefa 1. Testar as diversas funções que o sistema disponibiliza.

## 8.2. Calendarização

Lista de Milestones:

- Desenho do sistema
  - Entrega dos storyboards, desenhos do modelo do domínio e modelo de Entidade-Associação e o esboço da interface;
- Implementação da base de dados
  - Entrega de uma versão do software onde são criadas as tabelas da base de dados;
- Desenvolvimento das interfaces
  - Entrega dos ficheiros correspondentes á interface do utilizador, dos alunos, professor e dos administradores de sistema;
- Desenvolvimento da interação com o servidor
  - Entrega de uma nova versão do software onde foi realizado todo o tratamento e processamento dos dados relacionados com a interação com o servidor;
- Login
  - Entrega de uma nova versão do software onde é possível efetuar login através da aplicação;
- Funcionalidades dos administradores de sistema
  - Entrega de uma nova versão do software onde os requisitos funcionais requerentes aos administradores de sistema estão concluídos;
- Funcionalidades dos alunos
  - Entrega de uma nova versão do software onde os requisitos funcionais requerentes aos alunos estão concluídos;
- Funcionalidades dos docentes
  - Entrega de uma nova versão do software onde os requisitos funcionais requerentes aos docentes estão concluídos;
- Comunicação
  - Entrega de uma nova versão do software com um chat de troca de mensagens entre utilizadores;
- Implementação de segurança
  - Funcionalidades de segurança implementadas no sistema através de credenciais, criptografia e mecanismos de login para aceder ao servidor;
- Escalabilidade
  - Garantia de escalabilidade no sistema;
- Disponibilidade
  - Garantia de disponibilidade do serviço mesmo em alturas de maior fluxo de informação;
- Usabilidade
  - Entrega de uma versão do software onde as heurísticas e funções de usabilidade são garantidas na interface;

- Testes
  - Entrega de uma versão do software totalmente testada, sem erros e com todas as funcionalidades do sistema;
- Produto Final
  - Entrega do produto final.

Mapa de Gantt: Em anexo segue um ficheiro do Microsoft Project 2007 que contém o mapa de Gantt e a tabela calendarizada que contém o número de horas diárias de trabalho de cada elemento.

Nome do Recurso	Horas de trabalho
<b>André Silva</b>	314
<b>António Pinto</b>	338
<b>Diogo Frazão</b>	328
<b>Guilherme Nunes</b>	372
<b>João Pinto</b>	372
<b>Tiago Robalo</b>	348
<b>Vasco Bento</b>	324

*Tabela 11 – Número de horas de trabalho totais por recurso humano*

## **9. Gestão de riscos**

### **9.1. Lista de riscos**

- R-1. Falta de recursos
- R-2. Estabelecer prazos irrealistas
- R-3. Falta de competências necessárias na equipa
- R-4. Mau planeamento das tarefas
- R-5. Perda de versões e/ou documentos do projeto
- R-6. Falta de empenho de algum elemento do grupo
- R-7. Diferenças no acompanhamento do projeto por parte dos elementos
- R-8. Mudanças no planeamento a meio do desenvolvimento do projeto
- R-9. Desenho incorreto da aplicação
- R-10. Estruturação defeituosa da base de dados
- R-11. Desorganização do código
- R-12. Testes mal efetuados
- R-13. Ferramentas utilizadas não adequadas
- R-14. Desenvolvimento de funcionalidades desnecessárias
- R-15. O produto afasta-se dos requisitos do cliente



## 9.2. Tabela de riscos

Riscos	Categoria	Probabilidade	Impacto
R-13	Risco Técnico	Elevada	Elevado
R-3	Risco de Projeto	Média	Elevado
R-8	Risco de Projeto	Média	Elevado
R-1	Risco de Projeto	Média	Médio
R-2	Risco de Projeto	Média	Médio
R-4	Risco de Projeto	Média	Médio
R-6	Risco de Projeto	Média	Médio
R-7	Risco de Projeto	Média	Médio
R-12	Risco Técnico	Média	Médio
R-11	Risco Técnico	Média	Reduzido
R-5	Risco de Projeto	Reduzida	Elevado
R-15	Risco de Negócio	Reduzida	Elevado
R-9	Risco Técnico	Reduzida	Médio
R-10	Risco Técnico	Reduzida	Médio
R-14	Risco Técnico	Reduzida	Médio

*Tabela 12 – Tabela de riscos ordenada por probabilidade e impacto*

### **9.3. Plano RMMM**

#### **Risco 13 (Ferramentas utilizadas não adequadas):**

##### **Mitigação:**

1. Verificar se as ferramentas a ser utilizadas são adequadas para a realização do projeto.

##### **Monitorização:**

1. Acompanhar a utilização das ferramentas para o desenvolvimento do projeto.

##### **Gestão:**

1. Encontrar ferramentas adequadas para o desenvolvimento do projeto através de pesquisa e consulta do docente.

#### **Risco 3 (Falta de competências necessárias na equipa):**

##### **Mitigação:**

1. Realizar tutoriais online de modo a adquirir as competências necessárias, por ex.: CodeAcademy, YouTube, etc.
2. Reunir com restantes elementos da equipa e docentes para retirar dúvidas acerca das competências necessárias e nova tecnologia.

##### **Monitorização:**

1. Acompanhar a capacidade de resolução de problemas com as novas competências.

##### **Gestão:**

1. Encontrar soluções alternativas com competências já adquirias pelos membros da equipa.

#### **Risco 8 (Mudanças no planeamento a meio do desenvolvimento do projeto):**

##### **Mitigação:**

1. Seguir o planeamento feito na fase de planeamento do projeto.

##### **Monitorização:**

1. Verificar que o planeamento é seguido, de acordo com o que foi feito.

##### **Gestão:**

1. Reunir com os membros de equipa e explicar o acontecimento e corrigir.

## Conclusão

A execução deste relatório foi fundamental para planejar e prever a execução de uma aplicação que visa auxiliar alunos de um curso universitário na organização e gestão de grupos de trabalho. Deste modo, este relatório irá auxiliar o arranque, desenvolvimento e conclusão do projeto da cadeira de PTI e PTR a realizar no próximo semestre.

Este projeto será um grande desafio para todos os membros do nosso grupo, tal como na realização do relatório, encontrámos algumas fragilidades e incoerências que se irão transpor para o projeto. Apesar das adversidades encontradas, estas também têm um lado positivo, pois de certa forma servem para nos precaver de possíveis problemas que existam no decorrer do desenvolvimento da aplicação.

Para melhor sumarizar e retirar conclusões, iremos abordar nesta fase final, todos os pontos relevantes do relatório.

Nem todos os requisitos funcionais e não funcionais foram recolhidos à priori, com o decorrer do relatório, novas abordagens foram tidas em consideração, acrescentando as mesmas à lista, assim como alguns requisitos que perderam a sua viabilidade foram retirados. Determinados requisitos encontravam-se demasiado vagos ou complexos, e por forma a solucionar este problema decidimos dividi-los para que possam ser melhor compreendidos e contemplados de forma mais explícita no projeto.

Assim como nos requisitos acima referidos, também os dados de entrada e saída sofreram alterações no decorrer do relatório. Também foi pensado representar os dados de forma gráfica através de um diagrama, no entanto a sua representação não era explícita, dificultando a compreensão dos dados, logo, este não foi contemplado no relatório.

Relativamente aos recursos, nem sempre a perceção da disponibilidade de cada elemento do grupo correspondia com a realidade, e como tal a tabela 1 (*Tabela de Disponibilidade por Semana*) sofreu várias alterações até ser possível.

Em relação às ferramentas de desenvolvimento a nossa maior dificuldade surgiu na escolha da framework para o desenvolvimento do projeto, uma vez que a escolha desta é essencial para o bom funcionamento da aplicação. Após analisarmos as várias opções possíveis, percebemos que a que melhor se adequava às necessidades da aplicação e competências do grupo é o Laravel. Relativamente à sua linguagem, PHP está mais familiarizada e sendo uma framework MVC traduz-se numa mais valia organizacional, e em comparação a outras frameworks com a mesma linguagem, é a que possui mais documentação. Relativamente às outras opções tomadas, estas foram escolhidas por serem as mais adequadas à nossa abordagem ou por questões de familiaridade e quantidade de documentação.

A conclusão é especialmente relevante na secção 5. Com base nos valores obtidos, podemos afirmar que os valores referentes aos mesmos indicadores não estão em concordância uns com os outros, verificando-se discrepâncias nos valores obtidos referentes

aos mesmos indicadores com metodologias diferentes. Em 5.1, para aproximadamente 3 pessoas e 4 meses, o esforço total é de 11 PM.

Na secção 5.2, recorremos à recolha de dados históricos para fazer as estimativas, para tal, utilizamos um programa auxiliar “cloc” para nos auxiliar na contabilização das linhas de código de projetos passados. Com base na informação recolhida, foi obtido um valor para a produtividade de 2015 LOC/PM. Após a estimação das linhas de código previstas para o projeto, com base na decomposição do projeto em requisitos, obtivemos uma estimativa de 7223 LOC, posto isto foi nos possível calcular o esforço com base nas linhas de código e na produtividade obtida na secção anterior, o que culminou num esforço igual a 3,6 PM.

Por último, nas estimativas com base no modelo COCOMO, foi obtido um esforço de 26 PM, para uma duração de 9 meses e uma produtividade igual a 278 LOC/PM para um projeto com perto de 3 pessoas.

Alguns valores obtidos na secção 5 podem dar a impressão que a viabilidade do trabalho esteja comprometida, porém, após uma análise mais aprofundada aos valores obtidos e alguma discussão sobre os mesmo por parte dos elementos do grupo é possível concluir que à partida a execução do projeto será viável, dada a capacidade de trabalho de todos os elementos do grupo ser superior à que é tida em conta, bem como, por vezes o esforço obtido nas estimativas ter um valor deflacionado ou até mesmo termos em consideração uma equipa menor do que aquela que dispomos realmente.

O processo de desenvolvimento escolhido foi o Scrum, pois achamos ser o processo que melhor se adequava para o projeto e recursos que temos disponíveis. Por forma a auxiliar o processo de desenvolvimento, foi gerada a tabela presente na secção 7. Esta tabela gerou algumas dúvidas relativamente às tarefas, estas que foram mitigadas após uma sessão de dúvidas com o docente. Relativamente à secção 8, as dúvidas que mais persistiram na sua execução foram, em termos de organização, mais especificamente quanto ao rigor que era necessário na especificação de cada workpackage. Ainda na secção 8.2, as maiores adversidades encontradas foram a distribuição das horas e as precedências de cada tarefa.

Na secção da gestão de riscos, observamos que os riscos que tinham maior probabilidade poderiam vir a ter também um elevado impacto, isto através da linha de corte que separa os riscos com que o grupo se tem de preocupar durante o projeto. A linha de corte foi posicionada tendo em conta o Princípio de Pareto, adaptado ao risco, que enuncia que “80% do risco é causado por 20% dos riscos identificados”. Assim, os riscos acima da linha irão ser geridos usando o plano RMMM.

Mas, apesar de existirem riscos com alta probabilidade e alto impacto, a viabilidade do projeto não irá ser afetada, pois o grupo fará todos os possíveis para evitar estes riscos, em especial, o de utilização de ferramentas não adequadas, através da recolha extensa de informação e consulta do docente.

O mapa de Gantt veio ainda reforçar a viabilidade do projeto, pois foi possível estimar de uma forma mais pormenorizada a divisão de tarefas, e fazer uma estimativa do tempo despendido por cada elemento nas mesmas, e desta forma concluímos que com valores aceitáveis de trabalho por cada elemento será possível entregar o projeto antes da data prevista.

Outro fator que pode gerar alguns entraves no projeto das cadeiras de PTI e PTR é a parte do projeto relativa à segunda cadeira, pois no decorrer do curso, não foi dado grande ênfase à matéria relacionada com tecnologias de redes que achamos proveitosa no âmbito do desenvolvimento da aplicação. Também o facto da framework ser, para já, estranha aos elementos do grupo pode condicionar o arranque do projeto, porém será tido em conta um esforço inicial que irá compensar em fases posteriores.

Por último, um apoio fundamental que o grupo espera ter, para que o decorrer do projeto seja feito de forma menos atribulada, é o apoio e críticas feitas pelos professores nas reuniões mensais, que serão sempre encarados de forma construtiva e com o intuito de melhorar o produto final para que a aplicação seja concretizada com o maior sucesso possível.

## Bibliografia

- [1] Amazon.com, Inc., “Amazon Web Services, Inc.,” 2017. [Online]. Available: <https://aws.amazon.com/pt/ec2/>. [Acedido em Outubro 2019].
- [2] Oracle Corporation, “MySQL,” 2019. [Online]. Available: <https://www.mysql.com/>. [Acedido em Outubro 2019].
- [3] GitLab Inc., “GitLab,” Novembro 2019. [Online]. Available: <https://about.gitlab.com/>.
- [4] Cloud Native Computing Foundation, “Kubernetes,” 2019. [Online]. Available: <https://kubernetes.io/>. [Acedido em Novembro 2019].
- [5] JetBrains, “PhpStorm,” 2019. [Online]. Available: <https://www.jetbrains.com/phpstorm/>. [Acedido em Outubro 2019].
- [6] Google Inc., “Google Chrome,” 2019. [Online]. Available: <https://www.google.com/intl/pt-PT/chrome/>. [Acedido em Novembro 2019].
- [7] World Wide Web Consortium, “HTML5,” 2014. [Online]. Available: <https://html.spec.whatwg.org/multipage/>. [Acedido em Outubro 2019].
- [8] World Wide Web Consortium, “Cascading Style Sheets (CSS),” 2016. [Online]. Available: <https://www.w3.org/Style/CSS/>. [Acedido em Outubro 2019].
- [9] B. Eich, “JavaScript,” 2018. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Acedido em Outubro 2019].
- [10] The PHP Group, “PHP,” 2019. [Online]. Available: <https://www.php.net/>. [Acedido em Outubro 2019].
- [11] D. D. Chamberlin e R. F. Boyce, “SQL,” 2016. [Online]. Available: <https://www.w3schools.com/sql/>. [Acedido em Novembro 2019].
- [12] Bootstrap Core Team, “Bootstrap,” 2019. [Online]. Available: <https://getbootstrap.com/>. [Acedido em Outubro 2019].
- [13] JS Foundation, “jQuery,” 2018. [Online]. Available: <https://jquery.com/>. [Acedido em Outubro 2019].

- [14] World Wide Web Consortium, "AJAX," 2016. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>. [Acedido em Outubro 2019].
- [15] T. B. Otwell, "Laravel," 2019. [Online]. Available: <https://laravel.com/>. [Acedido em Novembro 2019].
- [16] Facebook Inc., "WhastApp," 2019. [Online]. Available: <https://www.whatsapp.com/>. [Acedido em Outubro 2019].
- [17] Discord, Inc, "Discord," 2019. [Online]. Available: <https://discordapp.com/>. [Acedido em Outubro 2019].
- [18] Microsoft Skype Division, "Skype," 2019. [Online]. Available: <https://www.skype.com/>. [Acedido em Outubro 2019].
- [19] R. S. Pressman e B. R. Maxim, Software Engineering : A Practitioner's Approach 8th Edition, New York: McGraw-Hill Education, 2014, pp. 744-753.