



**Ciências  
ULisboa**

Faculdade  
de Ciências  
da Universidade  
de Lisboa

# **Planeamento e Gestão de Projetos**

2017-2018

## **Relatório Final**

**Grupo 001**

**Autores:**

André Nunes, fc43304

Ana Catarina Sousa, fc48301

Hugo Filipe Curado, fc48761

Patrícia Jesus, fc46593

Pedro Duarte Neto, fc48758

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Âmbito do projeto</b>	<b>3</b>
2.1	Requisitos funcionais . . . . .	3
2.2	Requisitos não-funcionais . . . . .	4
2.3	Informação de entrada e saída . . . . .	5
<b>3</b>	<b>Planeamento</b>	<b>6</b>
3.1	Estimativas . . . . .	6
3.1.1	Esforço disponível . . . . .	6
3.1.2	Dados históricos . . . . .	6
3.1.3	Estimativa de linhas de código . . . . .	9
3.1.4	Estimativa COCOMO . . . . .	10
3.1.5	Análise crítica . . . . .	11
3.2	Recursos . . . . .	12
3.3	Processo de desenvolvimento de software . . . . .	14
3.4	Organização da equipa . . . . .	14
3.5	Planeamento do Projeto . . . . .	15
3.6	Gestão de Riscos . . . . .	15
<b>4</b>	<b>Conclusão</b>	<b>16</b>
<b>5</b>	<b>Bibliografia</b>	<b>17</b>

# 1. Introdução

O objectivo para este trabalho é planear e conceber uma plataforma de controlo de acessos da Faculdade de Ciências. Atualmente o acesso aos diversos espaços da faculdade (salas de aula, laboratórios, gabinetes, garagem, etc.) não é adequado às necessidades da comunidade escolar da faculdade (alunos, professores, funcionários, etc.) portanto foi-nos pedida uma solução que pudesse minimizar este problema, tornando o acesso mais prático, de fácil gestão e manutenção, mantendo todas as suas funcionalidades e segurança.

No sistema actual por vezes não é possível fazer um controlo rigoroso sobre a entrada de pessoas não autorizadas em espaços reservados, o que leva a uma visão irrealista e pouco segura.

Outro problema encontrado foi o controlo de pessoas num determinado espaço, como por exemplo o número de alunos numa determinada sala ou auditorio.

Num caso mais extremo de uma catástrofe, não existe uma visão de quantas pessoas estão num espaço que é necessário evacuar.

De modo a minimizar o problema do controlo de acessos, foi-nos proposto desenvolver uma solução tecnologica. Esta solução tem como base desenvolver um sistema acessível através de uma interface Web, disponível num ambiente "cloud" e de fácil integração com o hardware existente na faculdade. É necessário também que este sistema seja seguro e disponível. Devido à sensibilidade dos dados da aplicação, tais como nome dos alunos e professores, horários, o sistema deve usar mecanismos de segurança adequados de modo a manter a confidencialidade dos dados e a estar protegido contra vulnerabilidades e ataques.

Em relação à disponibilidade, o sistema deverá tolerar a falha de de um qualquer componente de hardware com uma redução mínima de desempenho e sem perda de dados, e tolerar uma falha catastrófica com um período de indisponibilidade não superior a 24h, sendo admissível apenas a perda de dados que tenham sido introduzidos no sistema nas últimas 24h.

A análise de requisitos, fontes de informação e tratamento da informação foram efectuadas na disciplina de Conceção de Produto no semestre passado, e é com base nessa informação recolhida que desenvolvemos esta solução.

Posto isto é necessário primeiro planificar quais os requisitos do sistema, as necessidades tecnológicas, riscos e organização da equipa. É também necessário escolher um modelo de desenvolvimento para o projecto e planear a execução do mesmo.

## 2. Âmbito do projeto

### 2.1 Requisitos funcionais

#### 1. Aluno

- (a) Consulta de histórico de acessos pessoal
- (b) Horário do curso frequentado
- (c) Horário dos espaços livres e a que grupo pertence (Ex.: grupo = Departamento de Informática)

#### 2. Funcionário de Departamento

- (a) Consulta de histórico de acessos pessoal
- (b) Consulta de salas disponíveis
- (c) Marcação de salas

#### 3. Secretaria

- (a) Consultar e alterar presenças nas aulas, mediante justificação válida

#### 4. Professor

- (a) Consulta dos alunos presentes nas suas aulas
- (b) Horário de aulas e "compromissos"
- (c) Consulta de histórico de acessos pessoal
- (d) Consulta de salas disponíveis
- (e) Marcação de salas para avaliações contínua e aulas extra

#### 5. Seguranças

- (a) Consulta de histórico de acessos
- (b) Consulta de salas disponíveis
- (c) Criar acesso para visitantes
- (d) Criar eventos de incidentes / desastres / perdidos e achados
- (e) Criar acessos temporários no caso de perda de cartão
- (f) Consulta de quantas pessoas estiveram presentes num edifício ou numa sala num determinado espaço de tempo
- (g) Consulta de quem esteve presente num edifício ou numa sala num determinado espaço de tempo, em caso de desastre, validado por mais que uma pessoa com um nível de Administrador de Sistema / Administrador-Chefe
- (h) Bloqueio de acessos em caso de necessidade, Lock Down
- (i) Desbloqueio do estado Lock Down, validado por mais que uma pessoa com um nível de Administrador de Sistema
- (j) Consultar acessos ao parque de estacionamento

## 6. Administrador

- (a) Consulta de quantas pessoas estiveram presentes num edifício ou numa sala num determinado espaço de tempo
- Consulta (b) **Registo** das salas disponíveis num determinado espaço temporal
- (c) Registo de novos utilizadores
- (d) Consulta de histórico de acessos pessoal
- (e) Alteração de estatutos, inferior ao seu nível de acesso
- (f) Associar cartão de aluno no sistema informático da dsi
- (g) Criar acesso para visitantes
- (h) Alterar acesso a salas e laboratórios
- (i) Consulta de salas disponíveis
- (j) Consultar acessos ao parque de estacionamento

## 7. Administrador-Chefe (Ex. Zé Fernandes)

- (a) Consulta de quem esteve presente num edifício ou numa sala num determinado espaço de tempo, em caso de um incidente
- (b) Consulta de quem esteve presente num edifício ou numa sala num determinado espaço de tempo, em caso de desastre, validado por mais que uma pessoa com um nível de Administrador de Sistema
- (c) Todas as funcionalidades do Administrador

## 8. Administrador de Sistemas

- (a) Alteração de todos os estatutos
- (b) Todas as funcionalidades do Administrador

## 2.2 Requisitos não-funcionais

1. Acessível através de interface web na cloud, compatível com qualquer browser e dispositivo.
2. Facilidade de integração com o hardware existente.
3. Necessidade de ligação constante à internet.
4. Confidencialidade dos dados.
5. **Um tipo de utilizador** terá acesso a uma visão global da base de dados em caso de necessidade (Ex.: Desastre natural ou roubo numa determinda sala)
6. Tolerar a falha de um qualquer componente de hardware com uma redução mínima de desempenho e sem perda de dados
7. Tolerar uma falha catastrófica com um período de indisponibilidade não superior a 24h, sendo admissível apenas a perda de dados que tenham sido introduzidos no sistema nas últimas 24h
8. Ser escalável e modular, por forma a suportar facilmente a adição e remoção de hardware para fazer face a picos de utilização que se prevê que ocorram em determinados momentos.
9. Integração no sistema de autenticação atualmente presente na faculdade.

## 2.3 Informação de entrada e saída

Dados recebidos e enviados para os diferentes serviços (lista)

### Entrada inicial de informação

Os dados sobre os utilizadores são gerados automaticamente por um serviço de simulação da interação de um grupo de pessoas com um edifício, sendo depois acedidos por **JSON**. Os dados simulados são dados referentes à implementação de um novo sistema de controlo de acessos. Este sistema é composto por aquisição do id do sítio por onde o utilizador passou, um timestamp que representa a hora a que passou e o id do utilizador. Estes pontos de controlo **estaram** inseridos antes e depois da porta ou entrada de qualquer espaço da faculdade. Sendo depois processados pela nossa aplicação. Para os dados relativos aos utilizadores, tais como nome, número da faculdade, curso, entre outros, é necessário haver uma ligação à base de dados académica da faculdade de modo a termos acesso a essa informação. Para a informação sobre horários, salas e espaços é necessário uma ligação ao sistema Fenix da Faculdade de Ciências. Para a informação sobre o estacionamento na faculdade é necessário uma ligação ao sistema Verex da faculdade.

em JSON

### Saídas de informação

Os dados de saída **seram** os dados relativos ao histórico de acesso de utilizadores, informação sobre a disponibilidade e ocupação dos espaços, incidentes e perdidos e achados, permissões de cada utilizador.

presenças serviços academicos

## 3. Planeamento

### 3.1 Estimativas

#### 3.1.1 Esforço disponível

Início do Projeto a 5 de Fevereiro com conclusão prevista a 7 de Julho, o que resulta num total de 5 meses.

Elementos da equipa e respetivas disponibilidades:

André = 35%	Patrícia = 40%
Catarina = 35%	Pedro = 35%
Hugo = 35%	

Tendo em conta a disponibilidade de cada elemento a equipa terá  $0.35 * 4 + 0.4 = 1.8$  pessoas. Com uma duração prevista de **5 meses** teremos um **esforço disponível** de  $1.8 * 5 = 9$  Pessoa Mes

#### 3.1.2 Dados históricos

Produtividade média USAR historico

**André, Catarina, Hugo e Pedro**

**Projeto de Programação I** em que dados 2 ficheiros texto era produzido um ficheiro com a informação pretendida, para tal os dois ficheiros eram lidos e manipulados em memoria. O projeto teve uma **duração aproximada de um mês** e cada grupo desenvolveu, aproximadamente **100 linhas de código** em **python**.

**Esforço** =  $2 * 0.2(1 \text{ cadeira em } 5) \text{ pessoas} * 1 \text{ mês} = 0.4 \text{ PM}$

**Produtividade** =  $100 \text{ LOC} / 0.4 \text{ PM} = 250 \text{ LOC/PM}$

**Projeto de Programação II** em que dado um ficheiro csv, eram produzidos gráficos com a informação mais relevante do ficheiro em causa. O projeto teve uma **duração aproximada de um mês** e cada grupo desenvolveu, aproximadamente **100 linhas de código** em **python**, destaque para o uso do modulo pylab.

**Esforço** =  $2 * 0.2(1 \text{ cadeira em } 5) \text{ pessoas} * 1 \text{ mês} = 0.4 \text{ PM}$

**Produtividade** =  $100 \text{ LOC} / 0.4 \text{ PM} = 250 \text{ LOC/PM}$

**Projeto de Introdução às Tecnologias Web** consistia em desenvolver um website a correr localmente só em browser. O tema era o jogo da forca. O projeto teve a **duração aproximada de dois meses e meio** e cada grupo desenvolveu aproximadamente **300 linhas de código** usando **HTML, CSS e Javascript**.

**Esforço** =  $3 * 0.2 \text{ pessoas} * 2.5 \text{ meses} = 1.5 \text{ P M}$

**Produtividade** =  $300 \text{ LOC} / 1.5 \text{ PM} = 200 \text{ LOC/PM}$

#### 3 mini-projetos de Sistemas Operativos

O **primeiro projeto** tratava de ficheiros duplicados em sistemas de ficheiros. Foi desenvolvido em **shell script**, teve **duração aproximada de três semanas** e **100 linhas de código**.

O **segundo mini-projeto** centrava-se em cifrar/decifrar ficheiros usando programação paralela. Teve uma **duração de três semanas**, desenvolvido em **python** e teve **200 linhas de código**.

**O terceiro mini-projeto** foi um aprimoramento do segundo em que se acrescentava a escrita em binário num ficheiro (log), se contabilizava o tempo de execução e se detetavam sinais (Ex.: SIGINT). Teve uma **duração de três semanas** e teve **aproximadamente 80 linhas de código**. Desenvolvido em **python**.

**Esforço** =  $3 \times 0.2 \text{ pessoas} \times 2 \text{ meses} = 1.2 \text{ PM}$

**Produtividade** =  $380 \text{ LOC} / 1.2 \text{ PM} = 317 \text{ LOC/PM}$

**Projeto de Sistemas Inteligentes** foi desenvolvido o jogo dos peões e respetivas funções de avaliação, teve **aproximadamente duração de 1 mês** e contou com **200 linhas de código** em **python**.

**Esforço** =  $3 \times 0.2 \text{ pessoas} \times 1 \text{ mes} = 0.6 \text{ PM}$

**Produtividade** =  $200 \text{ LOC} / 0.6 \text{ PM} = 333 \text{ LOC/PM}$

**André, Catarina, Hugo, Patrícia e Pedro**

**Projeto de Programação Centrada em Objetos** desenvolvido em duas fases, teve como objetivo o desenvolvimento uma plataforma de apoio a um restaurante, desenvolvido em **Java**. **200 linhas de código** e uma **duração de dois meses**.

**Esforço** =  $2 \times 0.2 \times 2 \text{ meses} = 0.8 \text{ PM}$

**Produtividade** =  $200 \text{ LOC} / 0.8 \text{ PM} = 250 \text{ LOC/PM}$

**Projeto de Bases de Dados** desenvolvido em duas fases, em que a primeira foi o desenho conceptual do enunciado e a segunda foi o mapeamento do desenho em SQL(DDL E DML). Teve uma **duração de 2 meses** e um **código de 250 linhas**.

**Esforço** =  $3 \times 0.2 \times 2 \text{ meses} = 1.2 \text{ PM}$

**Produtividade** =  $250 \text{ LOC} / 1.2 \text{ PM} = 208 \text{ LOC/PM}$

**4 mini-projetos de Aplicações Distribuídas** desenvolvidos ao longo do semestre.

**O primeiro projeto** foi o desenvolvimento de um servidor com recursos a serem disponibilizados, incluindo os respetivos locks, teve **duração de 3 semanas** e **aproximadamente 200 linhas de código**.

**O segundo projeto** foi um incremento do primeiro quanto às funcionalidades, tendo em conta a forma de comunicação e apresentação de conteúdos, teve a **duração de 2 semanas** e **aproximadamente 200 linhas de código**.

**O terceiro projeto** baseado num serviço WEB para um sistema simplificado, usou-se **sqlite3** e **flask**, teve duração de **3 semanas** e **aproximadamente 200 linhas de código**.

**O quarto projeto** foi um aprimoramento do terceiro e consistiu em fazer a comunicação segura do serviço WEB, HTTPS, chave publicas e privadas, teve **duração de 2 semanas** e **aproximadamente 100 linhas de código**.

**Esforço** =  $3 \times 0.2 \times 2.5 \text{ meses} = 1.5 \text{ PM}$

**Produtividade** =  $700 \text{ LOC} / 1.5 \text{ PM} = 467 \text{ LOC/PM}$

**Projeto de Analise e Desenho de Software** foi dividido em 2 fases, teve por objetivo o desenvolvimento de um sistema para uma cadeia de supermercados. Para tal foi utilizado o unified process. Na 1ª fase foi feita a analise e desenho. Na 2ª fase foi realizada a implementação e testes, teve **duração de 2 meses** e **200 linhas de código**, aproximadamente.

**Esforço** =  $3 \times 0.2 \times 2 \text{ meses} = 1.2 \text{ PM}$

**Produtividade** =  $200 \text{ LOC} / 1.2 \text{ PM} = 167 \text{ LOC/PM}$

**Projeto de Conceção do Produto** em que o tema é o mesmo que o da cadeira de Planeamento e Gestão de Projeto (PGP). , Neste projeto foi realizada a analise do problema a abordar em PGP e posteriormente no projeto final.



Catarina, Hugo e Pedro

**Projeto de Interação de Computadores** sobre uma aplicação tablet de encomenda de comida, ao longo do projeto foram desenvolvidos protótipos, questionários e por fim foi desenvolvido um protótipo funcional. Este projeto teve **duração aproximada de 2,5 meses** e um **código de 200 linhas**. Foi desenvolvido em **HTML, CSS e Javascript**.

**Esforço** =  $3 \times 0.2 \times 2.5 \text{ meses} = 1.5 \text{ PM}$

**Produtividade** =  $200 \text{ LOC} / 1.5 \text{ PM} = 133 \text{ LOC/PM}$

Patrícia

**Projeto de Interação de Computadores** sobre uma aplicação tablet de gestão de uma cozinha, ao longo do projeto foram desenvolvidos protótipos, questionários e por fim foi desenvolvido um protótipo funcional. Este projeto teve **duração aproximada de 2,5 meses** e um **código de 500 linhas**. Foi desenvolvido em **HTML, CSS e Javascript**.

**Esforço** =  $3 \times 0.2 \times 2.5 \text{ meses} = 1.5 \text{ PM}$

**Produtividade** =  $500 \text{ LOC} / 1.5 \text{ PM} = 333 \text{ LOC/PM}$

Catarina, Hugo e Pedro

**Projeto de Aplicações e Serviços Web** consistiu em desenvolver uma aplicação Full Stack para jogos de poker online, teve uma **duração de dois meses e meio**, cada projeto teve aproximadamente **500 linhas de código**. Para o desenvolvimento foram usadas as linguagens **php, javascript, html e css** e como framework **CodeIgniter**.

**Esforço** =  $3 \times 0.2 \times 2.5 \text{ meses} = 1.5 \text{ PM}$

**Produtividade** =  $500 \text{ LOC} / 1.5 \text{ PM} = 333 \text{ LOC/PM}$

André

**Projeto de Aplicações e Serviços Web** consistiu em desenvolver uma aplicação Full Stack para jogos de poker online, teve uma **duração de dois meses e meio**, teve aproximadamente **1500 linhas de código**. Para o desenvolvimento foram usadas as linguagens **javascript, html e css** com recurso ao MEAN Stack.

**Esforço** =  $3 \times 0.2 \times 2.5 \text{ meses} = 1.5 \text{ PM}$

**Produtividade** =  $1500 \text{ LOC} / 1.5 \text{ PM} = 1000 \text{ LOC/PM}$

Patrícia

**Projeto de Aplicações e Serviços Web** consistiu em desenvolver uma aplicação Full Stack para leilões, teve uma **duração de dois meses e meio**, cada projeto teve aproximadamente **6500 linhas de código**. Para o desenvolvimento foram usadas as linguagens **php, javascript, sql, html e css**.

**Esforço** =  $3 \times 0.2 \times 2.5 \text{ meses} = 1.5 \text{ PM}$

**Produtividade** =  $6500 \text{ LOC} / 1.5 \text{ PM} = 4333 \text{ LOC/PM}$

Pegar em projetos relevantes para o projeto e fazer média ponderada

### 3.1.3 Estimativa de linhas de código

Comparar linhas de código hist. com cocomo

Com base nos dados históricos de cada membro da equipa, foi feita a estimação das linhas de código nas diferentes perspectivas (otimista, provável e pessimista). De modo a ter uma melhor noção mais realista, dividimos o projecto nas várias componentes e estimamos a sua dimensão. As linhas de código para o simulador de dados não foram tidas em conta no cálculo final da ELC, isto porque os dados seriam obtidos usando hardware disponível na faculdade caso seja para produção.

	Otimista	Provável	Pessimista	Final
Angular / CLI				
Servidor MEAN	400	600	1000	633
Controlador	800	1000	1500	1050
Views	850	1200	1600	1208
Servidor LAMP	450	800	1200	808
Controlador	750	1000	1300	1008
Modelos	250	500	800	508
Base de dados	150	350	500	342
Segurança	150	500	800	492
Simulador de dados	350	500	800	525
Total	3800	5950	8700	6048

Falta contar

Tabela 3.1: Estimativa de Linhas de Código

$$EAL \text{ Final} = \frac{Otimista + 4 * Provavel + Pessimista}{6} = 6048$$

Estimativa demasiado pessimista

### 3.1.4 Estimativa COCOMO

Para este projeto optámos pelo modelo de COCOMO orgânico visto que:

- a compreensão do produto é alargada com base no conhecimento já obtido no projeto da cadeira de CP,
- a experiência em projetos semelhantes foi obtida ao longo da carreira académica de cada um dos elementos do grupo de trabalho,
- a necessidade de conformidade com pré requisitos é básica, visto que os requisitos funcionais do projeto não são muito extensos, focando-se em duas funcionalidades principais, registo de presenças e controlo de acessos,
- a necessidade de conformidade com interfaces externas é básica, visto que não nos foi dado acesso aos recursos já existentes, sendo estes dados simulados por um único recurso,
- a necessidade de desenvolvimento comcorrente de hardware e software não existe dado que não é o foco do projeto o desenvolvimento de hardware compatível com o sistema a desenvolver e o projeto não irá ser integrado com o sistema já existente na faculdade,
- a necessidade de estruturas de dados ou algoritmos inovadores pode ser considerada mínima visto que, apesar de ser necessário uma base de dados extensa para o backend não irá existir a necessidade algoritmos inovadores,
- o interesse em terminar o produto cedo é média visto que o projeto tem um prazo de entrega fixo com duração máxima de 5 meses,
- a dimensão do produto enquadra-se dentro do modo orgânico sendo previstas **2,75 KLOC**

6 KLOC

	Classificação				
	Muito reduzido	Reduzido	Nominal	Elevado	Muito elevado
<b>Atributos do produto</b>					
Fiabilidade do software necessária	.75	.88	1.0	1.15	1.4
Dimensão da base de dados		.94	1.0	1.08	1.16
Complexidade do produto	.70	.85	1.0	1.15	1.3
<b>Atributos do computador</b>					
Restrições ao tempo de execução			1.0	1.11	1.3
Restrições ao armazenamento de dados			1.0	1.06	1.21
Volatilidade da máquina virtual		.87	1.0	1.15	1.3
Tempo disponível do computador		.87	1.0	1.07	1.15
<b>Atributos dos indivíduos</b>					
Capacidade dos analistas	1.46	1.19	1.0	.86	.71
Experiência no desenv. De aplicações	1.29	1.13	1.0	.91	.82
Capacidade de programação	1.42	1.17	1.0	.86	.70
Experiência com a máquina virtual	1.21	1.1	1.0	.90	
Experiência com a linguagem de prog.	1.14	1.07	1.0	.95	
<b>Atributos do projecto</b>					
Uso de práticas modernas de programação	1.24	1.1	1.0	.91	.82
Uso de ferramentas de software	1.24	1.1	1.0	.91	.83
Prazos de desenvolvimento	1.23	1.08	1.0	1.04	1.1

Tabela 3.2: Tabela de valores de multiplicadores de esforço

Com base no COCOMO intermédio usando o modelo orgânico e tendo em conta os multiplicadores de esforço retirados da tabela dos Slides da TP06 [1] obtivemos os seguintes valores:

$$KLOC = 6048/1000 = 6.05$$

$$\text{Esforço [E]} = a * KLOC^b * EAF \Leftrightarrow 3.2 * 6.05^{1.05} * 0.87 = 18.43PM$$

$$\text{Duração [D]} = c * E^d * EAF \Leftrightarrow 2.5 * 10.80^{0.38} = 7.56Meses$$

$$\text{Numero de Pessoas [N]} = E/D \Leftrightarrow N = 18.43/7.56 = 2.43Pessoas$$

### 3.1.5 Análise crítica

Depois de uma análise aos dados históricos e a disponibilidade de cada elemento do grupo concluímos que o projecto terá uma duração prevista de **5 meses** e um esforço de **9 PM**. Com os valores finais estimados das linhas de código obtivemos um KLOC de **6048**, sem contar com a implementação do simulador de dados. De acordo com os cálculos do COCOMO intermédio obtivemos um esforço de **18.43 PM** e uma duração de **7.56 meses**.

Perante estes valores podemos observar que a duração é muito superior ao tempo disponível que temos para a elaboração do projecto. Uma das possíveis causas para esta discrepância pode ser devido ao facto de que a estimação das linhas de código ter-se baseado em projectos de pequena dimensão e em tecnologias e arquiteturas diferentes. Visto que o esforço necessário para o projecto é de **18.43 PM**, mais de metade do esforço disponível do grupo e a duração ter ultrapassado mais de 20% da duração disponível, decidimos aplicar o modelo intermédio do COCOMO a uma nova tabela de Estimativa de Linhas de Código. Esta nova tabela contempla todas as funcionalidades fundamentais pedidas no enunciado, garantindo assim que no final temos uma aplicação para apresentar.

	Otimista	Provável	Pessimista	Final
Servidor MEAN	150	250	500	275
Controlador	300	450	700	467
Views	300	500	800	517
Servidor LAMP	150	350	650	367
Controlador	350	450	800	492
Modelos	200	400	700	417
Base de dados	100	150	300	167
Segurança	100	200	500	233
Simulador de dados	350	500	800	525
Total	1650	2750	4950	<b>2935</b>

Tabela 3.3: Estimativa de Linhas de Código

$$\begin{aligned} \text{EAL Final} &= \frac{\text{Otimista} + 4 * \text{Provavel} + \text{Pessimista}}{6} = 2935 \text{ KLOC} = 2935/1000 = 2.935 \\ \text{Esforço [E]} &= a * KLOC^b * EAF^c \Leftrightarrow 3.2 * 2.935^{1.05} * 0.87 = 8.62PM \\ \text{Duração [D]} &= c * E^d * EAF \Leftrightarrow 2.5 * 10.80^{0.38} = 5.66Meses \\ \text{Numero de Pessoas [N]} &= E/D \Leftrightarrow N = 8.62/5.66 = 1.52Pessoas \end{aligned}$$

Com base nestes resultados obtivemos então um esforço de **8.62 PM** e uma duração de **5.66 meses**. Estes valores já são aceitáveis para a elaboração do projecto pois correspondem às previsões efectuadas da equipa. Na tabela acima estão descritas as linhas de código para realizar as funcionalidades necessárias para obter aprovação às disciplinas de PTI e PTR, incluindo o mínimo de segurança e fiabilidade do sistema.

Falta ligar com dados históricos

## 3.2 Recursos

Descrição	Disp. Média	Quando Necessário	Tempo Necessário
André Nunes	35%	80%	8:00 horas
Ana Catarina Sousa	35%	80%	8:00 horas
Hugo Filipe Curado	35%	80%	8:00 horas
Patrícia Jesus	40%	90%	8:00 horas
Pedro Duarte Neto	35%	80%	8:00 horas

Tabela 3.4: Tabela de Recursos Humanos

Pessoa / Semana	05-Feb	12-Feb	19-Feb	26-Feb	05-Mar	12-Mar	19-Mar
André Nunes	90,00%	90,00%	50,00%	50,00%	40,00%	20,00%	25,00%
Ana Catarina Sousa	70,00%	70,00%	50,00%	50,00%	40,00%	20,00%	20,00%
Hugo Filipe Curado	90,00%	90,00%	50,00%	50,00%	40,00%	20,00%	20,00%
Patricia Jesus	90,00%	90,00%	40,00%	30,00%	40,00%	55,00%	40,00%
Pedro Neto	70,00%	70,00%	50,00%	50,00%	35,00%	20,00%	20,00%

Pessoa / Semana	26-Mar	02-Apr	09-Apr	16-Apr	23-Apr	30-Apr	07-May
André Nunes	25,00%	20,00%	30,00%	30,00%	20,00%	30,00%	30,00%
Ana Catarina Sousa	25,00%	20,00%	40,00%	40,00%	20,00%	20,00%	50,00%
Hugo Filipe Curado	25,00%	20,00%	30,00%	30,00%	20,00%	30,00%	30,00%
Patricia Jesus	25,00%	50,00%	50,00%	45,00%	30,00%	50,00%	50,00%
Pedro Neto	30,00%	20,00%	40,00%	40,00%	20,00%	20,00%	50,00%

Pessoa / Semana	14-May	21-May	28-May	04-Jun	11-Jun	18-Jun	25-Jun
André Nunes	35,00%	30,00%	40,00%	40,00%	40,00%	50,00%	50,00%
Ana Catarina Sousa	20,00%	20,00%	40,00%	30,00%	30,00%	30,00%	30,00%
Hugo Filipe Curado	20,00%	30,00%	40,00%	40,00%	40,00%	50,00%	50,00%
Patricia Jesus	45,00%	30,00%	45,00%	30,00%	50,00%	30,00%	70,00%
Pedro Neto	20,00%	20,00%	40,00%	30,00%	30,00%	30,00%	30,00%

Pessoa / Semana	02-Jul
André Nunes	100,00%
Ana Catarina Sousa	80,00%
Hugo Filipe Curado	100,00%
Patricia Jesus	90,00%
Pedro Neto	80,00%

Planeamento por por semana

Tabela 3.5: Disponibilidade por pessoa por semana

	Modelos	Testes Modelos	Design	Testes Design	Construir e Executar testes	MEAN	LAMP	Segurança	Redes
Ana Sousa	E	E	E	E	E	B	M	M	M
André Nunes	M	M	M	M	E	E	B	M	E
Hugo Curado	M	M	M	M	E	M	E	E	E
Patricia Jesus	B	M	M	M	M	E	E	M	B
Pedro Neto	E	E	B	B	E	M	M	E	E

Tabela 3.6: Tabela de competências da equipa

Falta SoftSkills

Descrição	Disp.	Quando Necessário	Tempo Necessário	Categoria
Bootstrap	100%	Durante o desenvolvimento do front end	Duração do projeto	off-the-shelf
CodeIgniter / Laravel	100%	Para o back end	Duração do projeto	off-the-shelf
Microsoft SQL Server	100%	Para o back end	Duração do projeto	off-the-shelf
MySql	100%	Para o front end	Duração do projeto	off-the-shelf
jQuery	100%	Para o front end	Duração do projeto	off-the-shelf
Angular	100%	Para o back end	Duração do projeto	off-the-shelf
Express	100%	Para o back end	Duração do projeto	off-the-shelf
Docker / Kubernetes	100%	Durante o deployment do projeto		off-the-shelf

Tabela 3.7: Tabela de Recursos Software

Descrição	Disp.	Quando Necessário	Tempo Necessário
Servidor Local	500%	Durante o desenvolvimento do projeto em ambiente local	Duração do projeto
IDE de Desenvolvimento	500%	Durante todo o tempo de desenvolvimento	Duração do projeto
Browser	500%	Durante todo o tempo de desenvolvimento	Duração do projeto
Debugger do browser	500%	Durante todo o tempo de desenvolvimento	Duração do projeto
Postman ou similar	500%	Durante todo o tempo de desenvolvimento	Duração do projeto

Tabela 3.8: Tabela de Recursos Ferramentas

### 3.3 Processo de desenvolvimento de software

A nossa escolha sobre o modelo de processo a utilizar recai sobre o modelo incremental porque é aquele que melhor se adapta à equipa e ao produto a desenvolver.

Visto ser um modelo constituído por mini cascatas conseguimos ter uma funcionalidade a 100% mais rapidamente do que com qualquer outro modelo de processo. Isto será especialmente relevante nas reuniões com os clientes (professores), pois, assim conseguimos apresentar funcionalidades com maturidade de software e assim o retorno será mais preciso, ao contrário do que acontece com outros modelos de processo em que as funcionalidades nem sempre apresentam maturidade de software o que pode levar a um retorno mais vago.

### 3.4 Organização da equipa

O modelo de organização da equipa a usar é o modelo matricial. Esta escolha justifica-se principalmente pela modularidade do projecto em causa, o que elimina a organização horizontal, o facto de ser descentralizado é um factor a favor e elimina o modelo hierárquico que é centralizado.

Este modelo de organização da equipa funciona bem com o modelo de processo do projecto porque ambos se focam na modularidade.

	Especialista de Front-End	Architecture Designer	Especialista em Redes	Especialista em Segurança	Especialista de Back-End	Especialista em Testing	Analista	Gestor de Projeto	Cliente
Análise de Front-End	C						RA	I	C
Design de Front-End	R							I	
Implementação de Front-End	RA	RA						I	
Testing de Front-End	R							I	
Análise de Back-End						RA	RA	I	C I
Design de Back-End		RA			C			I	
Implementação de Back-End					R			I	
Testing de Back-End					RA			I	
Análise de Segurança				C		RA	RA	I	CI
Design de Segurança		RA		R				I	C
Implementação de Segurança				RA				I	
Testing de Segurança				R		RA		I	
Análise de Simulador de Dados							RA	I	CI
Design de Simulador de Dados		RA						I	C
Implementação de Simulador de Dados								I	
Testing de Simulador de Dados						RA		I	
Análise de Redes			C				RA	I	CI
Design de Redes		RA	R					I	C
Implementação de Redes			RA					I	
Testing de Redes			R			RA		I	CI

Tabela 3.9: Tabela RACI

	Ana Sousa	André Nunes	Hugo Curado	Patricia Jesus	Pedro Neto
Especialista Front-End	X	X	X		
Especialista Back-End		X	X	X	
Architecture Designer	X				X
Especialista Testing	X			X	X
Especialista Segurança		X	X		X
Especialista Redes	X	X			X
Analista	X			X	
Gestor de Projeto		X			

Tabela 3.10: Tabela de competências

Falta tarefas do gestor e com demasiado trabalho

### 3.5 Planeamento do Projeto

Por favor consultar o ficheiro planeamentoPGP001.mpp.

### 3.6 Gestão de Riscos

#	Risco	Prob.	Impacto	Categoria
1	Equipa não familiarizada com tecnologias	4	Projeto: 2, Produto: 4	Projeto
2	Má estimacão da complexidade	4	Projeto: 4, Produto: 3	Projeto
3	Incompatibilidade entre as tecnologias	3	Projeto: 4, Produto: 4	Projeto
4	Requisito mal entendido	3	Projeto: 4, Produto: 4	Projeto
5	Software não apresenta maturidade (bugs)	3	Projeto: 3, Produto: 4	Técnico
6	Sobrecarga trabalhos (outras disciplinas)	3	Projeto: 3, Produto: 3	Projeto
7	Má configuração dos servidores	2	Projeto: 5, Produto: 5	Técnico
8	Prometer demasiado aos clientes(professores)	2	Projeto: 4, Produto: 4	Negócio
9	Elemento da Equipa adoeecer	2	Projeto: 3, Produto: 3	Projeto

Tabela 3.11: Tabela de Gestão de Risco

#	Mitigação	Monitorização	Gestão
1	Dividir equipa em cada competência (Tabelas de Recursos); Documentação	Acompanhar projetos e desenvolvimento de cada componente	Rodar pessoas, reunir e explicar caso haja alguém fluente
2	Tendo em conta a estimacão pessimista faz se uma maior divisão de componentes; Alargar prazo da componente	Acompanhar desenvolvimento de cada componente; Verificar atrasos; Acompanhar dificuldades no desenvolvimento	Colocar mais pessoas a trabalhar na componente

Tabela 3.12: Tabela RMMM

Critério de org.



## 4. Conclusão

Tendo como base a análise feita na cadeira de Conceção de Produto e tendo analisado um pouco mais a arquitetura estrutural das necessidades para a elaboração do projecto elaboramos este planeamento. A nossa solução foi pensada de modo a simplificar as interações dos utilizadores com os espaços da faculdade e ao mesmo tempo mantendo todo o controlo e segurança necessários. Com os primeiros valores do COCOMO podemos concluir que o projecto planeado é demasiado ambicioso para o tempo disponível. Perante o nosso modelo de desenvolvimento incremental, decidimos refazer os cálculos de modo a elaborarmos primeiramente os requisitos fundamentais para obter aprovação nas cadeiras de PTI e PTR. Assim sendo concluimos que o projecto é viável se ser executado no tempo pretendido.

Contudo, o grupo compromete-se a apresentar uma solução baseada nas funcionalidades descritas nos requisitos funcionais do projecto, dando prioridade às funcionalidades core. Com o trabalho realizado ao longo desta cadeira, reconhecemos um dos principais obstáculos à elaboração de um projecto, ou seja, um dos factores mais importantes é ter um bom planeamento que corresponda com as necessidades técnicas, estruturais e tecnológicas do mesmo. Muitas das vezes a parte do planeamento pouco elaborada e por causa disso a execução do projecto pode ser a esperada.

Muitas das vezes a parte do planeamento é pouco elaborada e por causa disso a execução do projecto pode não ser a esperada.

Tirar milestones dentro das iterações

Review

Falta Deliverables

Total de Horas

Trabalho por semana

Trabalho a fazer no project nao plan inicial

Segurança bootstrap pouco tempo

## 5. Bibliografia

- [1] C. Duarte, *pgp 1718 - aula 6 - estimação.pdf*, [https://moodle.ciencias.ulisboa.pt/pluginfile.php/15607/mod\\_resource/content/3/pgp%201718%20-%20aula%206%20-%20estima%20o.pdf](https://moodle.ciencias.ulisboa.pt/pluginfile.php/15607/mod_resource/content/3/pgp%201718%20-%20aula%206%20-%20estima%20o.pdf), Departamento de Informática, FCUL, Outubro 2017.