



**Ciências
ULisboa**

Faculdade
de Ciências
da Universidade
de Lisboa

Planeamento e Gestão de Projetos

2017-2018

Relatório FASE I

Grupo 001

Autores:

André Nunes, fc43304

Ana Catarina Sousa, fc48301

Hugo Filipe Curado, fc48761

Patrícia Jesus, fc46593

Pedro Duarte Neto, fc48758

Conteúdo

1	Introdução	2
2	Âmbito do projeto	3
2.1	Requisitos funcionais	3
2.2	Requisitos não-funcionais	4
2.3	Informação de entrada e saída	5
3	Planeamento	6
3.1	Estimativas	6
3.1.1	Esforço disponível	6
3.1.2	Dados históricos	6
3.1.3	Estimativa de linhas de código	8
3.1.4	Estimativa COCOMO	8
3.1.5	Análise crítica	8
3.2	Recursos	8
3.3	Processo de desenvolvimento de software	8
3.4	Organização da equipa	8
3.5	Planeamento do Projeto	9
3.6	Gestão de Riscos	9
4	Conclusão	10
5	Bibliografia	11

1. Introdução

Este projeto tem como a objetivo criar uma plataforma de armazenamento, consulta e gestão de informação relativa aos acessos aos edifícios e salas da Faculdade de Ciências da Universidade de Lisboa, tendo em conta o hardware existente no espaço em causa e as necessidades do mesmo.

Esta plataforma será disponibilizada numa solução híbrida, integrando recursos em cloud (AWS) e recursos locais.

A necessidade de efetuar um levantamento rigoroso das atuais soluções apresentadas aos utilizadores foi um dos trabalhos realizados no âmbito da cadeira de Conceção do Produto no ano letivo 2016/2017. Com este objetivo alcançado conseguimos ganhar algum tempo sendo que agora só será necessário melhorar alguns aspetos.

Em relação ao projeto em causa, no mínimo, terão que ser sustentadas três funcionalidades, uma para cada grupo de utilizadores:

1. Aos administradores da plataforma, consulta de quem esteve presente num edifício ou numa sala em determinados períodos temporais
2. Aos professores da FCUL, consulta dos alunos presentes nas suas aulas
3. Aos alunos da FCUL, consulta do seu histórico de acessos

Passivamente o sistema sustentará, no mínimo, os seguintes requisitos:

1. Tolerar a falha de um qualquer componente de hardware com uma redução mínima de desempenho e sem perda de dados;
2. 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;
3. 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.
4. confidencialidade de dados, tendo em conta a sensibilidade dos dados a manipular.

2. Âmbito do projeto

2.1 Requisitos funcionais

1. Funcionário de Limpeza / Manutenção
 - (a) Consulta de histórico de acessos.
 - (b) Consulta de salas disponiveis.
2. Aluno
 - (a) Consulta de histórico de acessos.
 - (b) Horário do curso frequentado.
 - (c) Horário dos espaços, de acordo com o currículo.
 - (d) Espaços livres a determinadas horas e a que grupo pertence.
3. Funcionário de Departamento
 - (a) Consulta de histórico de acessos.
 - (b) Consulta de salas disponiveis.
 - (c) Marcação de salas.
4. Secretaria
 - (a) Consultar e alterar preseças nas aulas.
 - (b) Alterar premissão de acesso ao parque de estacionamento.
5. Professor
 - (a) Consulta dos alunos presentes nas suas aulas.
 - (b) Horário de aulas e "compromissos".
 - (c) Consulta de histórico de acessos.
 - (d) Consulta de salas disponiveis.
 - (e) Marcação de salas.
 - (f) Reserva de sala para exame.
6. Seguranças
 - (a) Consulta de quem esteve presente num edifício ou numa sala num determinado espaço de tempo, em caso de um incidente.
 - (b) Consulta de histórico de acessos.
 - (c) Consulta de salas disponiveis.
 - (d) Criar acesso para visitantes.
 - (e) Criar eventos de incidentes / desastres / perdidos e achados.
 - (f) Criar acessos temporários no caso de perda de cartão.

- (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.

7. Administrador

- (a) Consulta de quantas pessoas estiveram presentes num edifício ou numa sala num determinado espaço de tempo.
- (b) Registo das salas disponiveis em tempo real.
- (c) Registo de novos utilizadores.
- (d) Alteração de estatutos, inferior ao seu nível de acesso.
- (e) Associar cartão de aluno no sistema informático da dsi.
- (f) Criar acesso para visitantes.
- (g) Alterar acesso a salas e laboratórios.

8. Administrador-Chefe (Ex. Ze 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 histórico de acessos.
- (c) Consulta de salas disponiveis.
- (d) Criar acesso para visitantes.
- (e) 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.
- (f) Consultar acessos ao parque de estacionamento.

9. Administrador de Sistemas

- (a) Acesso à API.
- (b) Alteração de todos os estatutos.
- (c) 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. Usar protocolos que sustentem os requisitos funcionais e não funcionais.
6. Cada utilizador só tem acesso a uma vista da base de dados coerente com o seu nível de acesso.
7. Nenhum utilizador terá acesso a uma visão global da base de dados.
8. Tolerar a falha de um qualquer componente de hardware com uma redução mínima de desempenho e sem perda de dados
9. 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

10. 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.
11. Execução de um sistema de simulação de dados de entrada sobre os utilizadores do sistema, salas, edificios e parque de estacionamento.
12. Integração no sistema de autenticação atualmente presente na faculdade.

2.3 Informação de entrada e saída

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 edificio, sendo depois acedidos por JSON. Dados sobre marcação de salas, exames, incidentes.

Saídas de informação

Toda a informação necessária para responder as necessidades dos requisitos funcionais.

3. Planeamento

3.1 Estimativas

3.1.1 Esforço disponível

Elementos da equipa e respectivas disponibilidades:

André = 35%	Patricia = 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 $1.8*5=9$ Pessoas/Mês

3.1.2 Dados históricos

André, Catarina, Hugo e Pedro

Projecto 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 projecto 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 = $100 \text{ LOC} / 0.4 \text{ PM} = 250 \text{ LOC/PM}$

Projecto de Programação II em que dado um ficheiro csv, eram produzidos gráficos com a informação mais relevante do ficheiro em causa. O projecto 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 = $380 \text{ LOC} / 1.2 \text{ PM} = 317 \text{ LOC/PM}$

Projecto de Sistemas Inteligentes foi desenvolvido o jogo dos peões e respectivas 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 Objectos 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 respectivos 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 Projecto (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 = $200 \text{ LOC} / 1.5 \text{ PM} = 133 \text{ LOC/PM}$

Patricia

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 projecto teve aproximadamente **500 linhas de código**. Para o desenvolvimeto 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 **500 linhas de código**. Para o desenvolvimeto 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 = $500 \text{ LOC} / 1.5 \text{ PM} = 333 \text{ LOC/PM}$

Patricia

Projeto de Aplicações e Serviços Web consistiu em desenvolver uma aplicação Full Stack para leilões, teve uma **duração de dois meses e meio**, cada projecto teve aproximadamente **6500 linhas de código**. Para o desenvolvimeto 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 = $6500 \text{ LOC} / 1.5 \text{ PM} = 4333 \text{ LOC/PM}$

3.1.3 Estimativa de linhas de código

3.1.4 Estimativa COCOMO

3.1.5 Análise crítica

3.2 Recursos

3.3 Processo de desenvolvimento de software

2ª entrega

3.4 Organização da equipa

2ª entrega

3.5 Planeamento do Projeto

2^a entrega

3.6 Gestão de Riscos

4. Conclusão

5. Bibliografia