

Planeamento e Gestão de Projecto

Relatório Fase 2

Alexandre Machado, nº 43551

Nuno Silva, nº 44285

Francisco Pires, nº 44314

11 de Dezembro de 2015

Conteúdo

1	Introdução	3
2	Análise de requisitos	4
2.1	Requisitos funcionais e não funcionais	4
2.1.1	Requisitos funcionais	4
2.1.2	Requisitos não funcionais	5
2.2	Modelo de casos de uso	7
2.2.1	Casos de uso textuais	7
2.2.2	Diagrama modelos de uso	10
2.3	Esboço das interfaces	10
2.4	Modelo de dados e requisitos detalhados	10
3	Planeamento	11
3.1	Recursos	11
3.1.1	Recursos Humanos	11
3.1.2	Disponibilidade	11
3.1.3	Organização da equipa	12
3.1.4	Tabela de competências	12
3.1.5	Justificações	13
3.1.6	Recursos de hardware e de software	14
3.2	Estimação	15
3.2.1	Esforço disponível	15
3.2.2	Linhas de código	15
3.2.3	Modelos Empíricos	16
3.3	Processo de Desenvolvimento de Software	16
3.4	Gestão de Riscos	18
3.4.1	Riscos de Recursos Humanos	19
3.4.2	Riscos de Tecnologia	19
3.4.3	Riscos no desenvolvimento do projecto	20
3.4.4	RMMM	20
4	Arquitetura	23
5	Conclusão	25

Capítulo 1

Introdução

Foram feitas alterações nas partes entregues na primeira e segunda fase

Este projecto tem como objectivo o desenvolvimento e a implementação de um Sistema de Informação (SI), dirigido aos utentes do Serviço Nacional de Saúde (SNS). Este relatório vai abordar a análise dos requisitos, o planeamento e por ultimo a arquitetura que vai ser necessária para a realização deste projecto.

O objectivo final vai ser construir um SI mais eficaz e seguro, que agregue mais funcionalidades que o actual. Para isso, foi feito um levantamento extensivo das funcionalidade que podem ser introduzidas no Portal de Utente, sem que estas sejam inconsistentes com os restantes serviços disponibilizados pelo SNS. Para alem disso, foi também feita um levantamento das melhores tecnologias a utilizar neste tipo de projectos, para poder cumprir todos os requisitos não funcionais que vão ser referidos mais a frente no relatório.

Pretendemos também com a realização do planeamento e do SI familiarizar-nos com as técnicas e tecnologias a serem utilizadas no mercado de trabalho.

Capítulo 2

Análise de requisitos

2.1 Requisitos funcionais e não funcionais

2.1.1 Requisitos funcionais

De acordo com os objectivos definidos para este projecto, seleccionamos as seguintes funcionalidades que o utilizador terá disponíveis neste SI.

- Registo de contactos e dados pessoais
 - N° Cartão de Cidadão, N° Utente de Saúde, N° telefone, N° Identificação Fiscal, Email
- Definir agregado familiar
 - Pai, mãe, filhos, irmãos, etc...
- Identificação de cuidador familiar
- Registo de informação pessoal relevante
 - Testamento Vital, Contados de emergência
- Registo de indicadores básicos de saúde
 - Medicação, alergias, diagnósticos, cirurgias, vacinação, doenças raras
- Registo de exames complementares de diagnóstico
 - Todo o tipo de exames(documentos relevantes) que o utente faça podem ser guardados no portal pelo próprio
- Consulta de registos clínicos
- Pedido de prescrição de medicação crónica

- No caso da medicação crónica, o portal pode gerar receitas automaticamente mediante as instruções do medico.
- Marcação de consultas
- Inscrição e consulta das listas para cirurgia (eSIGIC)
- Definir estado no Registo Nacional de Não Dadores (RENDA)
- Pesquisa de serviços médicos (directório)
 - Poder pesquisar serviços médicos por área
- Pedido de mudança de médico de família
- Pedido de isenção de taxas moderadoras

2.1.2 Requisitos não funcionais

Para execução das funcionalidades neste SI, será necessário assegurar os requisitos não funcionais que listamos de seguida.

- Confidencialidade dos dados
 - Alguns dados médicos que o utente introduz na plataforma só vão poder ser acedidos pelo seu médico mediante autorização do utente.
- Segurança dos dados e dos acessos
 - O acesso aos dados de um utilizador específico vai somente ser permitido a esse utilizador e aos seus médicos. Para aceder a esses dados deve-se:
 - * Como utilizador - autenticar a sua conta
 - * Como médico - autenticar a sua conta e esperar que o sistema automaticamente dê as devidas permissões de acesso aos dados do utilizador.
- Garantia de disponibilidade
 - Garantir que a plataforma está sempre acessível online (24/7).
- Escalável e modular
 - Capacidade de poder aumentar a capacidade de servir um maior numero de clientes, iniciando paralelamente mais instâncias do servidor de aplicação e de *web*.
- Tempo de resposta
 - A plataforma terá um tempo de resposta inferior a 500ms.

- Assegurar o cumprimento das normas legais do SNS
- Resolução de conflitos
 - Garantir que a informação que o utente e o médico introduzem não divergem, fazendo *merge* de todos os dados introduzidos e pedir aos utilizadores para confirmarem as alterações feitas.
- Persistência, sincronização dos dados e Disponibilidade
 - A informação guardada nos vários servidores vai ser distribuída por várias instâncias. Estas instâncias vão ser sincronizados automaticamente para não haver falhas de persistência e de sincronização, e para além disso, vai permitir uma rápida recuperação para o SI poder tolerar falhas de hardware.
- Notificações e alertas de acontecimentos do utilizador
 - Estas serão enviadas para o utente ou o medico via email ou telemóvel
- *Responsive Web Design*
 - Garantir que o site pode ser acedido por vários tipos de dispositivos (*e.x: tablets, smartphones, etc*)

2.2 Modelo de casos de uso

2.2.1 Casos de uso textuais

Consultar registos clínicos

Ator Principal: Utilizador.

Interesses: O utilizador pretende consultar os seus registos clínicos.

Pré-condições: O utilizador está registado na plataforma.

Pós-condições: O utilizador consiga visualizar os seus registos clínicos com sucesso.

Cenário principal de sucesso:

1. O utilizador indica que quer consultar os seus registos clínicos.
2. O sistema mostra todos os registos existentes sobre o utilizador.

Cenários alternativos:

- 2.b. O utilizador não tem informação registada nos seus registos.
- 3.b. O sistema notifica o utilizador que ainda não registou os seus registos médicos.
- 4.b. O sistema dá oportunidade para o utilizador completar os seus registos médicos.
- 5.b. O utilizador escreve os seus registos médicos.

Marcar cirurgias

Ator Principal: Utilizador.

Interesses: O utilizador pretende marcar uma cirurgia.

Pré-condições: Utilizador está autorizado a marcar a cirurgia por um elemento medico.

Pós-condições: Utilizador consegue marcar uma cirurgia numa certa data com sucesso.

Cenário principal de sucesso:

1. O utilizador indica que quer marcar uma cirurgia.
2. O sistema notifica ao utilizador que foi autorizado a marcar uma cirurgia de um certo tipo.
3. O sistema mostra as datas disponíveis pelo sistema para a marcação.

4. O utilizador marca a cirurgia numa das datas disponíveis.

Cenários alternativos:

- 2.b. O utilizador não tem autorização para marcar a cirurgia.
- 3.b. O sistema informa o utilizador que este não tem autorização para marcar a cirurgia pretendida.
- 4.b. O sistema informa o utilizador para se dirigir a um posto medico para receber a tal autorização.

Registar exames

Ator Principal: Utilizador.

Interesses: Utilizador pretende marcar um exame.

Pré-condições: Utilizador está autorizado a marcar o exame por parte de um elemento médico.

Pós-condições: Utilizador consegue marcar um exame numa certa data com sucesso.

Cenário principal de sucesso:

1. O utilizador indica que quer marcar uma exame.
2. O sistema notifica ao utilizador que foi autorizado a marcar um exame de um certo tipo.
3. O sistema mostra as datas disponíveis pelo sistema para a marcação.
4. O utilizador marca o exame numa das datas disponíveis.

Cenários alternativos:

- 2.b. O utilizador não tem autorização para marcar o exame.
- 3.b. O sistema informa o utilizador que este não tem autorização para marcar o exame pretendido.
- 4.b. O sistema informa o utilizador para se dirigir a um posto medico para receber a tal autorização.

Pedir medicamentos

Ator Principal: Utilizador.

Interesses: O utilizador pretende renovar os seus medicamentos.

Pré-condições: A receita é renovável e está dentro da data limite desta (seis meses).

Pré-condições: A receita é passada pelo medico de família do utilizador.

Pós-condições: O utilizador consegue renovar a sua dose do medicamento em questão.

Cenário principal de sucesso:

1. O utilizador navega até à pagina de renovações de medicação.
2. O sistema informa que utilizador têm uma receita renovável.
3. O utilizador escolhe a receita pretendida.
4. O utilizador imprime a receita para levar à farmácia.

Cenários alternativos:

- 2.b. O sistema informa o utilizador que este tem uma receita prescrita por um medico de família.
- 3.b. O sistema informa o utilizador que para renovar a medicação desta receita, o utilizador têm que se dirigir ao medico em questão para uma nova consulta.
- 2.c. O sistema informa o utilizador que este não têm nenhuma medicação que possa ser renovada.

Marcar consulta

Ator Principal: Utilizador.

Interesses: O utilizador pretende marcar uma consulta com um profissional de saúde.

Pré-condições: Ter medico de família e estar registado no portal.

Pós-condições: O utilizador consegue marcar uma consulta com um profissional de saúde com sucesso.

Cenário principal de sucesso:

1. O utilizador navega até à secção de marcação de consultas.
2. O sistema verifica se o utilizador têm um medico de família.
3. O sistema mostra datas disponíveis para o medico de família do utilizador
4. O utilizador marca uma consulta conforme os horários mostrados.
5. O sistema informa o utilizador que a sua consulta está em estado de aprovação por parte do medico.

6. O sistema informa o medico que um utilizador quer marcar uma consulta para aquela data.
7. O medico escolhe aceitar ou não aceitar a consulta.
 - 8.a. O utilizador recebe uma notificação com a aprovação da consulta feita pelo medico.
 - 8.b. O utilizador recebe uma notificação a dizer que o medico não aceita a consulta marcada.

Cenários alternativos:

- 3.b. Se não têm medico de família:
- 4.b. O sistema mostra as datas para a marcação de uma consulta externa.
- 5.b. O utilizador marca uma consulta para um medico disponível no centro de saúde mais próximo.
- 6.b. O sistema informa o utilizador que a sua consulta está em fase de aprovação pelo medico escolhido.
- 7.b. O sistema informa o medico que um utilizador quer marcar uma consulta para aquela data.
- 8.b. O medico escolhe aceitar ou não aceitar a consulta.
- 9.b.a. O utilizador recebe uma notificação com a aprovação da consulta feita pelo medico.
- 9.b.b. O utilizador recebe uma notificação a dizer que o medico não aceita a consulta marcada.

2.2.2 Diagrama modelos de uso

Em anexo, o ficheiro *DiagramaModelosDeUso.png* encontra-se o diagrama representado os Modelos de Uso do Utilizador e do Médico.

2.3 Esboço das interfaces

Em anexo, na pasta *Interface* encontra-se vários **.png* com os vários esboços das interfaces.

2.4 Modelo de dados e requisitos detalhados

Em anexo, o ficheiro *ModeloDeDados.png* encontra-se o modelo de dados.

Capítulo 3

Planeamento

3.1 Recursos

3.1.1 Recursos Humanos

Os recursos humanos para o projecto incluem seis alunos de Tecnologias de Informação (LTI), sendo que os três alunos não presentes neste relatório pertencem ao grupo 003. No final da cadeira de Planeamento e Gestão do Projecto (PGP), os dois grupos irão juntar-se e trabalhar em conjunto nas cadeiras de Projecto Tecnologias de Informação (PTI) e Projecto Tecnologias de Redes (PTR). A duração total do projecto será de sete meses, sendo três meses e meio dedicados ao planeamento (PGP).

3.1.2 Disponibilidade

A disponibilidade dos alunos é conforme apresentada na seguinte tabela:

	Disponibilidade	
	1ºSemestre	2ºSemestre
Pedro Neves	20%	40%
Rita Capela	20%	28,6%
Tiago Maurício	20%	28,6%
Francisco Pires	20%	33,3%
Alexandre Machado	20%	28,6%
Nuno Silva ¹	10%	*

Tabela 3.1: Tabela de Disponibilidade

¹O aluno em questão encontra-se a trabalhar em *part-time*, pelo que no primeiro semestre tem menos disponibilidade. Não sendo possível prever, por agora, a sua disponibilidade no segundo semestre, foi decidido não ser calculada.

3.1.3 Organização da equipa

A organização dos membros envolvidos vai ser feita em três grupos, utilizando um modelo horizontal (democrática e descentralizada). Um grupo para PTR, um para PTI, e um ultimo grupo para os "*elementos moveis*". Estes alunos vão contribuir em conjunto para o trabalho de ambas as cadeiras, e ao mesmo tempo, gerir o funcionamento e as decisões dos grupos.

- Grupo PTR
 - Francisco Pires
 - Nuno Silva
- Grupo PTI
 - Tiago Maurício
 - Rita Capela
- *Elementos Moveis*
 - Alexandre Machado
 - Pedro Neves

3.1.4 Tabela de competências

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pedro Neves	3	4	3	2	4	4	1	4	3	2	4	4	4	4	3
Rita Capela	3	2	4	4	3	4	4	4	4	3	1	4	4	3	4
Tiago Maurício	3	4	4	3	4	4	2	3	3	2	3	4	4	4	4
Francisco Pires	2	3	4	3	4	3	3	4	4	2	3	4	4	4	3
Alexandre Machado	2	4	4	4	4	4	3	4	4	3	3	3	4	3	4
Nuno Silva	2	4	4	4	4	4	3	3	3	2	4	4	4	4	3

Tabela 3.2: Tabela de Competências

1. PHP
2. Java
3. HTML
4. CSS
5. Python

6. MySQL
7. Interface
8. Gestão
9. Analise
10. Desenho
11. Testes
12. Comunicação
13. Trabalho em equipa
14. Trabalho sobre pressão
15. Capacidade em cumprir prazos

3.1.5 Justificações

A utilização de dois gestores num único projecto é uma organização de equipa excecional. Esta decisão foi tomada tendo em conta as características únicas deste grupo. Os membros de PTI e PTR foram escolhidos baseados nas suas competências técnicas, sendo que isto criou um conflito, pois os dois restantes membros consideram ter ambos competências técnicas equivalentes nas duas cadeiras. Isto, a juntar ao facto destes dois membros serem bastante superiores em termos de gestão de trabalho, levou-nos a tomar esta decisão incomum. Em termos práticos, estes dois elementos vão gerir equipas diferentes, só tendo de tomar decisões juntos quando influenciar todos os elementos ou projecto num todo.

3.1.6 Recursos de hardware e de software

Alguns destes pontos vão ser mais detalhados no capítulo 4.

Recursos de software

- Sistema Operativo: Ubuntu Server 14.4.
- Servidor Web: WildFly (existe a possibilidade de usar outro servidor baseado em Apache).
- Uma base de dados relacional MySQL.
- Java, versão 8 (utilizando as API de JEE para desenvolver o servidor aplicativo).
- IDE de Java (IntelliJ IDEA ou Eclipse)

Recursos de hardware

- Servidores da Amazon (ou outro servidor *cloud-based*)
- Computadores da ADMIN-FCUL para testes e servidores locais

3.2 Estimação

Para a realização da tabela relativa aos dados históricos, foram escolhidas as cadeiras em que a matéria dos projectos se encaixa no âmbito do projecto.

	AD	ASW	ITW	ADS	SO
Alexandre Machado	1002/160h	2576/160h	756/72h	454/18h	560/42h
Francisco Pires	1002/160h	NA	687/5h	NA	775/50h
Nuno Silva	942/150h	NA	542/10h	500/35h	700/60h

Tabela 3.3: Dados Históricos (*Lines of Code* e horas).

3.2.1 Esforço disponível

- 1º semestre (duração: 3,5 meses)

$$20 + 20 + 20 + 20 + 20 + 10 = 110 \text{ (1,1 pessoas)} \quad (3.1)$$

$$E = 1,1 \cdot 3,5 = 3,85 \text{ PM} \quad (3.2)$$

- 2º semestre (duração: 3,5 meses)

$$40 + 28,6 + 28,6 + 33,3 + 28,6 = 188 \text{ (1,88 pessoas)} \quad (3.3)$$

$$E = 1,88 \cdot 3,5 = 6,58 \text{ PM} \quad (3.4)$$

3.2.2 Linhas de código

As linhas de código previstas para o projecto são conforme apresentadas na seguinte tabela:

	Optimista	Provável	Pessimista	Final
Criar a Base de Dados	50	120	200	123
Configurar <i>HTTP Server</i>	5	20	50	25
Ligação à Base de Dados ²	5	10	20	12
Segurança	200	300	350	283
Sistema Distribuído ³	2000	3750	5000	3583
<i>Views</i>	1000	1500	2500	1600
Controlador	500	750	1000	750
Modelo	200	300	500	333
Total	3960	6750	9620	6777

Tabela 3.4: Linhas de Código

²Linhas a não serem consideradas usando a linguagem *Java*.

³Considera-se por SD a programação integral de um Sistema Distribuído. Caso se use um serviço que somente precise de configuração (p.ex. *Amazon Web Services*), estas linhas devem ser alteradas.

3.2.3 Modelos Empíricos

Calculo do esforço orgânico:

$$E = a . KLOC^b \quad (3.5)$$

$$E = 2,4 \left(\frac{N.Linhas}{1000} \right)^{1.05} \quad (3.6)$$

$$E = 2,4 \left(\frac{6777}{1000} \right)^{1.05} = 17.89 P.M \quad (3.7)$$

Calculo da Duração:

$$D = c . E^d \quad (3.8)$$

$$D = 2,5 (17,89)^{0,38} = 7,48 M \quad (3.9)$$

3.3 Processo de Desenvolvimento de Software

Como processo de desenvolvimento do nosso projeto decidimos usar o Processo Unificado. Esta decisão foi baseada numa reflexão da nossa parte, em que, pensámos na forma como trabalhamos e, visto que este projeto não é de forma alguma *full-time*, tivemos de ter isso em conta. O Processo Unificado permite-nos avançar iterativamente e ao mesmo tempo voltar a trás sem que hajam muitos problemas, havendo assim um balanço entre o avançar no projeto e ajustar problemas anteriores, o que achamos que seria perfeito no nosso caso.

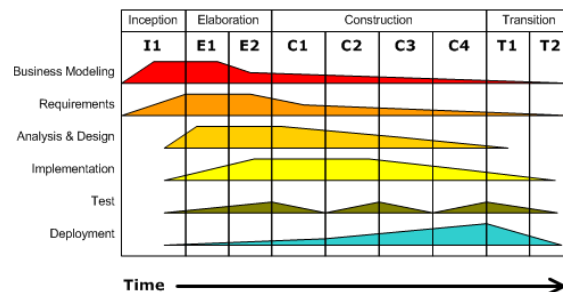


Figura 3.1: Exemplo de um Processo Unificado

O Processo Unificado divide-se em quatro fases:

1. *Inception* – justifica-se a execução do projeto, ou seja, tenta-se adquirir um conhecimento do que irá ser preciso para concluir o projeto e quando concluído, os resultados deste.
2. *Elaboration* – conclui-se de certa forma a fase de *inception*, visitando com mais detalhe todos os fatores de risco, *reward* e recursos que este irá trazer. Convém ser o mais completo e detalhado possível visto que na fase seguinte vai proceder-se à construção do projeto.
3. *Construction* – começa-se a construção do que irá ser uma versão operacional do projeto. O foco principal nesta fase é a construção de features discutidas anteriormente. É de valor notar que em projetos de maior dimensão esta fase poderá ter varias iterações.
4. *Transition* – o foco nesta fase será transitar o projeto de um ambiente de desenvolvimento para um ambiente de produção, pondo o produto disponível ao cliente final, para que este o perceba e o use. Nesta fase faz-se o treino do cliente final e o beta testing para validar o projeto em relação às expectativas do cliente final. De seguida compara-se o estado do projeto nesta fase à fase de *Inception* e se tudo estiver bem, faz-se uma *release*.

Vantagens do Processo Unificado:

- O cliente não precisa de esperar muito tempo para entrar em contacto com um resultado prático.
- Quando terminado o desenvolvimento do projeto é muito difícil encontrar erros dada a facilidade de os corrigir anteriormente.
- Os riscos de grau mais elevado são trabalhados em primeiro lugar, dando assim alguma confiança no desenvolvimento do projeto

Desvantagens do Processo Unificado:

- Poderá haver desorganização em períodos mais avançados no projeto.
- Aumento de gastos em implementações de varias versões do projetos.

3.4 Gestão de Riscos

Nesta avaliação dos riscos para o nosso projeto, identificámos que existem três grandes áreas: a de Relações Humanas, a de Tecnologia e a de Desenvolvimento do projeto.

Na categoria de RH identificámos que os principais problemas têm a haver com a relação entre membros do grupo e o comportamento de cada um.

Um outro facto de grande impacto que se prevê ser uma fonte de conflitos é a existência de dois gestores. Tendo já justificado esta organização de equipa incomum, os membros do projecto estão cientes que este tipo de organização pode complicar a tomada de decisões e atrasar o projecto, fazendo deste facto um risco de prioridade média.

Na categoria de Tecnologia identificámos que bugs e a segurança são os principais riscos a ter em conta e, iremos dar ênfase à segurança no projeto, visto que uma das partes mais críticas é o tipo de informação que iremos tratar.

Por ultimo, na categoria de desenvolvimento do projeto, identificámos que, sem surpresa, o maior problema são os atrasos que poderão acontecer, podendo estragar planos e horários planeados para a completção do projeto.

Em geral achamos que os nossos riscos irão ser de natureza comum a todos os grupos, são riscos que a maioria dos projetos, quer a nível académico ou profissional, encontram, não significando que os podemos levar menos a serio, sendo esta a causa de projetos falhados em varias áreas.

3.4.1 Riscos de Recursos Humanos

1. Má comunicação entre os elementos do grupo.
2. Falta de empenho de um ou mais elementos do grupo.
3. Falta de conhecimento em geral ou numa área específica.
4. Baixa de um membro do grupo.
5. Falta de *química* entre membros do grupo.
6. Atraso na entrega de trabalho de um membro do grupo.
7. Desistência de um membro do grupo.
8. Dois gestores

3.4.2 Riscos de Tecnologia

- 9 *Spaghetti code* - a partir de um ponto mais avançado no projeto poderá haver desorganização do código efetuado.
- 10 Código inútil proveniente de más práticas de desenvolvimento.
- 11 Má implementação (bugs) de uma funcionalidade.
- 12 Demora na descoberta de uma resolução para um bug encontrado na aplicação.
- 13 Demora no *patching* da aplicação quando esta sofre uma falha a nível de segurança.
- 14 *Updates* que pioram o uso ou a performance de SI.
- 15 Nova vulnerabilidade de segurança devido a update realizado.
- 16 Falta de segurança do projetos: intrusão externa à plataforma sem autorização, acesso indevido à base de dados, uso incorreto (inseguro) da plataforma.
- 17 Má performance da plataforma em tempos de maior tráfego devido a má configuração.

3.4.3 Riscos no desenvolvimento do projecto

- 18 Atrasos na entrega do projeto
- 19 Falta de funcionalidades na entrega do projeto final - Na entrega do projeto final ficamos aquém das expectativas que foram impostas no planeamento do projeto.
- 20 Requisitos incompletos - Não identificação de todos os requisitos essenciais para o bom funcionamento da plataforma em tempo de desenvolvimento.

Riscos	Tipo	Probabilidade	Impacto
1	Recursos Humanos	Baixa	2
2	Recursos Humanos	Média	2
3	Recursos Humanos	Média	3
4	Recursos Humanos	Baixa	3
5	Recursos Humanos	Média	3
6	Recursos Humanos	Média	2
7	Recursos Humanos	Baixa	3
7	Recursos Humanos	Média	3
8	Tecnologia	Elevada	2
9	Tecnologia	Baixa	2
10	Tecnologia	Elevada	2
11	Tecnologia	Média	2
12	Tecnologia	Média	3
13	Tecnologia	Baixa	2
14	Tecnologia	Média	3
15	Tecnologia	Elevada	3
16	Tecnologia	Média	3
17	Desenvolvimento	Elevada	1
18	Desenvolvimento	Média	2
19	Desenvolvimento	Baixa	1

Tabela 3.5: Tabela de Riscos

3.4.4 RMMM

A solução de riscos considerados elevados utilizando a RMMM segue-se abaixo.

Spaghetti code

Mitigação: Para não haver código desorganizado, o que pode levar a tempos excessivos de desenvolvimento, teremos de ter em mente as melhores práticas no

contexto em que estamos a trabalhar, neste caso Java.

Monitorização: Um elemento do grupo fazer uma inspeção periódica (a decidir mais tarde pelo manager) do código feito por outro elemento do grupo.

Gestão: Caso se verifique que estamos a perder eficiência a desenvolver o projeto devido à desorganização de código, tirar um tempo só para tentar organizar o projeto.

Má implementação (*bugs*) de uma funcionalidade

Mitigação: Tentar testar o máximo de casos possíveis relacionados com a funcionalidade em questão e ter em atenção na construção dessa mesma.

Monitorização: Ter em atenção ao feedback dos utilizadores para perceber se a funcionalidade foi bem implementada.

Gestão: Caso uma funcionalidade seja *deployed* e esteja mal implementada, tentar perceber o que está mal o mais rápido possível para que se possa arranjar o errado. Em caso crítico e, se possível, reverter a parte afetada para um estado anterior que se tenha a certeza que está correto.

Falta de segurança na implementação

Mitigação: Usar uma mentalidade de segurança em primeiro lugar quando na fase de desenvolvimento, visto estarmos a lidar com informação sensível e importante. Fazer um plano próprio para implementação de um sistema de segurança. Testar extensivamente. Definir configurações de firewall (IPTables, etc...) para mitigar grande parte do risco associado.

Monitorização: Ter em atenção a possíveis ataques que possam estar a ocorrer. Usar ferramentas próprias de monitorização de ataques maliciosos ao projeto (CloudFlare).

Gestão: No caso de algum problema de segurança encontrado, tentar perceber a gravidade e a natureza de tal. A partir desse ponto arranjar uma solução que pareça adequada, desde um simples patch à parte afetada ao caso mais extremo, um encerramento da aplicação.

Atrasos na entrega do projeto

Mitigação: Ter sempre em atenção as próximas etapas de entrega. Comunicação constante entre o grupo e colaboração máxima para tentar que não exista este problema.

Monitorização: Tentar perceber se nos estamos a atrasar o mais cedo possível para então se isso estiver a acontecer poder-mos ter alguma mão de manobra para arranjar solução para que não aconteça.

Gestão: No caso de nos atrasar-mos numa entrega, teremos de compensar na próxima. Melhor comunicação e colaboração entre o grupo será necessário. Alocação de uma carga horária maior.

Dois gestores do projecto

Mitigação: Os dois membros do grupo devem sempre tomar decisões juntos, e estar abertos a alterações, para assim criar menos conflitos na gestão do projecto. Em caso de conflito, devem discutir com os restantes membros do projecto, de maneira a haver uma decisão maioritária.

Monitorização: Os dois gestores devem manter-se sempre sincronizados, para as decisões serem consistentes e não haver conflito.

Gestão: Em caso de conflito, os gestores devem discutir com os restantes membros do projecto, de maneira a chegarem a um consenso.

Capítulo 4

Arquitetura

O SI vai ser dividido em quatro componentes de *software* principais.

1. Servidor *web*
2. Servidor aplicacional
3. Servidor de base de dados
4. Sistema operativo

1. Servidor *web*

O servidor *web* vai processar as paginas *web* para serem entregues ao cliente. As ligações vão ser protegidas com o protocolo SSL (criando paginas https). Vai ser também configurado uma *firewall iptables* para limitar os ip e portos de acesso.

2. Servidor aplicacional

O servidor aplicacional vai servir de *middleware* entre o servidor de base de dados e o servidor *web*. Esta *framework* vai ser programada usando o Java Applications Servers (JEE) e o *WildFly*. Vai também ser auxiliada por uma API da Twilio, para autenticação de dois passos por SMS.

3. Servidor de base de dados

O servidor de base de dados vai instanciar uma base de dados MySQL, criada a partir do JEE.

4. Sistema Operativo

O Sistema Operativo escolhido para a realização do projecto foi o Ubuntu Server 14.04.

Estes quatro componentes de software vão ser colocados em servidores *cloud*. Neste relatório vamos falar dos serviços da Amazon (AWS), mas deixamos em aberto usar outro serviço que oferecer mais vantagens ou que fique mais barato.

Amazon Web Services

O servidor *web* e o servidor aplicacional vão utilizar o serviço Amazon EC2 (um conjunto de servidores de computação escalável e flexível) e o servidor de base de dados vai utilizar o serviço Amazon RDS, que instância uma base de dados relacional para ser utilizada por uma aplicação *web* carregada nos servidores EC2.

Estes serviços, devidamente configurados, vão servir de sistema distribuído para o nosso SI, assegurando todos os requisitos não funcionais, e aumentando a disponibilidade e a eficácia do SI (em comparação com usar um rede configurada na FCUL).

Um Diagrama da interação com o cliente encontra-se no ficheiro *Arquitetura.png* em anexo.

Capítulo 5

Conclusão

Perante o projecto que nos foi proposto, definimos os requisitos funcionais e não funcionais como pilares da nossa proposta de trabalho. Através de uma pesquisa ao *website* do Portal do Utente e um conjunto de boas práticas de serviços *web*, adicionamos funcionalidades possíveis de implementar no SI, e que determinam uma melhoria, tanto no serviço, como na interação com o utilizador.

Bibliografia

- [1] Leslie Lamport *LaTEX: a document preparation system*, 2nd edition, 1994.
- [2] Roger S- Pressman, Bruce Maxim, *Software Engineering: A pratitioner's Approach*, McGraw-Hill, 8ª edição, 1973.
- [3] Página Web do Portal de Saúde <https://www.portaldasaude.pt/portal>, 2015. (Plataforma Dados Saúde)
- [4] Página Web do Portal de Utente <https://servicos.min-saude.pt/utente/>, 2015. (Plataforma Dados Saúde)
- [5] Página Web sobre Python <https://docs.python.org/2/>, 2015. (Documentação Python 2.7).
- [6] Página Web sobre Java <https://docs.oracle.com/javase/8>, 2015 (Documentação Java SE 8).
- [7] Amazon Web Service <https://aws.amazon.com/>, 2015.
- [8] Let's Encrypt <https://letsencrypt.org/>, 2015. (Certificados SSL)