

Instituto Politécnico de Setúbal



Escola Superior de Tecnologia de Setúbal

Software de Gestão para Escolas de Condução

João Portas	202002475
Lucas Freixieiro	202002193

Gestão de Projetos Informáticos
Licenciatura em Engenharia de Informática

Docente: Cédric Grueau

Setúbal, 30 de janeiro de 2023

Índice

1	Introdução	3
2	Âmbito do projeto	3
2.1	Nome do projeto.....	4
2.2	Resultado do projeto	4
2.3	Exclusões	5
3	Gestão das partes interessadas (stakeholders)	6
3.1	Stakeholders	6
3.2	Planos de comunicações	6
3.3	Matriz de responsabilidades RACI	7
4	Planeamento de recursos e custos	8
4.1	Recursos humanos por fase	8
4.2	Custos com espaço	10
4.3	Recursos informáticos	10
4.3.1	Digitais	10
4.3.2	Materiais	10
5	Planeamento do projeto	11
5.1	Estrutura Analítica do Trabalho (WBS)	11
5.2	Cronogramas do projeto	12
5.2.1	Estimativas de trabalho (WBS)	12
5.2.2	Critical Path Method (CPM).....	12
5.2.3	Diagrama de Gantt	13
6	Plano de qualidade	14
7	Planeamento e gestão de riscos	16
7.1	Riscos.....	16
7.2	Matriz de riscos	17
7.3	Plano de mitigação	17
8	Conclusão	20
9	Webgrafia	20

1 Introdução

Este documento contém diversos conteúdos para descrever o planeamento de um projeto no âmbito da unidade curricular de Gestão de Projetos Informáticos. Para projeto o tema escolhido foi fazer um planeamento para um software de gestão para escolas de condução.

2 Âmbito do projeto

Este projeto foi escolhido após perceber-se durante a pandemia do COVID-19 um problema em muitas escolas de condução. Por experiência nas escolas em que nós frequentamos e pelo que se teve feedback de outras pessoas recentemente com o mesmo problema.

O problema identificado foi a falta de facilidade nos pagamentos das cartas, exames, consultas de aulas e marcação de aulas. Pois, algumas destas teriam de ser tratadas presencialmente ou por telefone no caso das marcações de aulas, pelo que o principal problema era mesmo quando queríamos saber quantas aulas teóricas tínhamos ou teríamos de ter um papel com isso apontado ou as atendentes nas escolas tinham de contar todas as vezes que assinamos presença nas aulas.

Apesar de algumas escolas do país (Portugal) possuírem algum tipo de software, muitas delas são escolas em regiões mais populosas como Lisboa, e que muitas das vezes são mais dedicadas a registos internos e não tão pensadas para os alunos também.

Então, o que se pretende aqui é o desenvolvimento de um software que será um website que tem uma parte de gestão e estatísticas acessível a pessoas da administração, como o dono das escolas de condução, outra parte dedicada a funcionários e instrutores com funcionalidades que se dediquem a cada um dependendo dos seus papeis na escola e a parte principal do projeto que seria ter uma parte acessível ao aluno onde eles poderiam fazer a sua gestão e consultas de forma automática e independente.

2.1 Nome do projeto

Nomeou-se o nome do projeto como “OfficeDrive” que também será o nome do website.

2.2 Resultado do projeto

Este projeto será desenvolvido para web em que deverá ser construído sobre as tecnologias:

- Backend
 - Nodejs e Express
- Frontend
 - HTML, CSS, JavaScript, MaterialUI e React
- Base de dados
 - PostgreSQL

Terá de ter as seguintes funcionalidades:

- Permitir o gestor das escolas ter acesso às informações de todas elas.
- Permitir que os instrutores possam consultar as aulas marcadas de forma fácil.
- Permitir contabilizar de forma automática a quantidade de aulas que foram frequentadas pelos alunos tanto teóricas como práticas e disponibilizar para cada aluno esses dados.
- Cada utilizador deverá ter acesso aos dados no website de acordo com as suas permissões.
- Permitir o pagamento de prestações da carta online.
- Permitir a gestão de funcionários, alunos e veículos quanto ao seu registo na plataforma.
- Permitir a marcação de aulas na plataforma.
- Diferenciação de diferentes cartas nas marcações e gestão no caso de os alunos estarem a tirar diversas categorias.

2.3 Exclusões

Este projeto não deverá incluir os seguintes aspetos:

- Conexão com as finanças para declarar pagamentos feitos pelos alunos na plataforma pelo que deverão ser declarados posteriormente.
- Marcação e pagamento de exames, que por ser regulado por uma entidade externa deverão ser tratados diretamente na escola de condução.
- Sistemas de GPS ou algo parecido para controlar o decorrer das aulas de condução.
- Fazer testes teóricos na plataforma.
- Partilha de comentários, ideias ou conteúdos.

3 Gestão das partes interessadas (stakeholders)

3.1 Stakeholders

- **Dono da escola de condução:** O qual será importante mostrar o quão bom seria o nosso software, útil não só para os alunos, mas também para ele mesmo e como poderia melhorar a reputação e trazer maior lucro a longo prazo.
- **Atendentes das escolas (Secretária):** Que costumam ser quem tem de fazer todas as marcações das aulas e controlo de forma manual, dando mais trabalho e podendo produzir erros humanos. Estes deverão ter um maior interesse na parte que toca à marcação de aulas ao balcão e pedidos por parte dos alunos relativamente à obtenção das contagens das aulas e outras informações reduzindo assim a carga de trabalho relativamente a tarefas que podem ser automatizadas pelo software.
- **Instrutores:** Podem consultar também as suas aulas na plataforma de forma prática.
- **Alunos (instruendos):** Poderão fazer pagamentos da carta na plataforma, consultar, marcar aulas e outras coisas que antes teria de telefonar para a escola ou deslocar-se.
- **Equipa de desenvolvimento:** Irão fazer o desenvolvimento da aplicação pelo que será necessário ter um orçamento, um plano e cronogramas bem estruturados para incentivar o desenvolvimento desta plataforma.
- **Project Manager:** Responsável por planear e acompanhar o desenvolvimento da plataforma.
- **Equipa de marketing:** Tem a responsabilidade de gerir a satisfação dos clientes.

3.2 Planos de comunicações

Para pôr a par de tudo o que está a acontecer durante o desenvolvimento serão feitos “daily meetings” de 10 a 15 minutos com todas as partes que estão envolvidas dependendo de cada fase do projeto para fazer um ponto de situação do progresso. Estas reuniões poderão acontecer de forma remota através do Microsoft Teams ou Zoom.

Também serão feitas reuniões mais longas antes do início de cada fase para perceber o que será feito ao longo de cada uma delas.

3.3 Matriz de responsabilidades RACI

Table 1 - Matriz RACI

Posição	Recolha de requisitos	Documento de Requisitos,	Documento Técnico com a	Mockups de baixa fidelidade	Mockups de Alta fidelidade	Estudo de Usabilidade	Preparar Infraestrutura	Base de dados	Implementação (backend)	Implementação (frontend)	Documento de testes	Preparar ambiente de testes	Testes Unitários	Testes de integração	QA Testing	UAT	Deployment	Manutenção
Gestor de Projeto	R	A	A	I	I	A	A	A	A	A	A	A	I	I	I	I	A	A
Engenheiro DevOps	C	R	R				R	I	I	I	C	R		C	C	C	R	R
UX Designer	I	I	C	R	R	R				C	C					C	I	C
Admin de Base de Dados	I	I	C				C	R	C		C	C				C	I	R
Programador backend	C	C	I						R		I		R	R	R	R	I	R
Programador backend		I	I						R		I		R	R	R	R	I	
Programador frontend		I	I	I	I	I				R	I		R	R	R	R	I	R
Programador frontend		I	I	I	I	I				R	I		R	R	R	R	I	
Especialista em testes	I	I	C	I	I	C	I	I	I	I	R	C	A	A	A	A	C	C
Cliente	A			A	A				I	I						I	I	I
Legenda: R=Responsible; A=Accountable; C=Consulted; I=Informed																		

4 Planeamento de recursos e custos

4.1 Recursos humanos por fase

Para o desenvolvimento da aplicação irá ser necessário arranjar recursos humanos nas seguintes áreas:

- **Gestor de projeto:** Para recolher informações, identificar os requisitos e módulos e fazer o planeamento mais preciso da aplicação. Também irá acompanhar o projeto.
- **Programador:** Que saiba preparar e instalar servidores para fazer a parte dos servidores de teste para o de produção.
- **Engenheiro DevOps:** Para preparar a infraestrutura e tornar parte do processo automático.
- **UX designer:** Para fazer as mockups e fazer os estudos das mesmas quanto à sua usabilidade e outros estudos.
- **Programador backend:** Para desenvolver o código referente à lógica dos processos de negócio necessários para o funcionamento da aplicação na parte do servidor.
- **Programador frontend:** Para desenvolver a página web baseada nas mockups desenvolvidas.
- **Especialista de base de dados:** Para desenhar e desenvolver a base de dados.
- **Especialista em teste:** Para verificar se tudo funciona como deveria e está de acordo com o que foi projetado.

Table 2 – Recursos humanos por fase

Tarefas	Posição	Custo diário	Duração estimada (dias)	Custo estimado
Levantamento de requisitos			9	646,38 €
Entrevistas e análise	Gestor de projeto	71,82 €	9	646,38 €
Desenho			55	3 093,21 €
Documento alto nível	Gestor de projeto	71,82 €	7	502,74 €
	Programador	41,00 €	7	287,00 €
	Especialista de base de dados	55,27 €	7	386,89 €
	UX designer	53,87 €	17	915,79 €
	UX designer	58,87 €	17	1 000,79 €
Desenvolvimento			172	12 487,84 €
Acompanhamento do projeto	Gestor de projeto	71,82 €	45	3 231,90 €
	Engenheiro Devops	83,06 €	45	3 737,70 €
Preparação de servidor	Programador	41,00 €	2	82,00 €
Base de dados	Especialista de base de dados	55,27 €	12	663,24 €
Backend (servidor)	Programador backend	79,17 €	22	1 741,74 €
	Programador backend	79,17 €	22	1 741,74 €
Frontend (website interface)	Programador frontend	53,73 €	12	644,76 €
	Programador frontend	53,73 €	12	644,76 €
Testes				2 150,24 €
Acompanhamento do projeto	Engenheiro Devops	83,06 €	14	1 162,84 €
Documento de casos de teste	Especialista de testes	56,27 €	2	112,54 €
Preparação de servidor	Programador	41,00 €	2	82,00 €
Testes unitários	Programador backend	79,17 €	2	158,34 €
Testes de integração	Especialista de testes	56,27 €	5	281,35 €
Testes de qualidade	Especialista de testes	56,27 €	5	281,35 €
Testes de aceitação	Gestor de projeto	71,82 €	1	71,82 €
Instalação				271,16 €
Preparar aplicação	Programador	41,00 €	1	41,00 €
Instalar aplicação no servidor	Programador backend	79,17 €	2	158,34 €
Testes de aceitação	Gestor de projeto	71,82 €	1	71,82 €
Total				18 648,83 €

Após o término do projeto será necessário contratar também ou manter algumas áreas com salários mensais para a manutenção da plataforma. Os contratos e custos estão na tabela abaixo.

Table 3 - Custos de Manutenção

Posição	Qtn.	Salário Mensal	Total
Programador fullstack	1	1 600,00 €	1 600,00 €
Especialista de base de dados	1	1 600,00 €	1 600,00 €
Suporte de TI	3	831,00 €	2 493,00 €
Total			5 693,00 €

4.2 Custos com espaço

O trabalho seria principalmente remoto, mas para as reuniões presenciais iria-se ter um escritório arrendado com localização no Porto em Portugal.

Sendo que o escritório de 19 m² e tem um custo de 250€ por mês. Como o projeto deverá durar um de 4 meses daria um custo de 250€x4, tendo-se assim um custo total com o espaço de 1 000€.

4.3 Recursos informáticos

4.3.1 Digitais

Para o desenvolvimento e preservar os direitos para poder comercializar a aplicação foi feito o seguinte orçamento de preços para as licenças necessárias que estão na tabela abaixo.

Table 4 - Recurso digitais (licenças)

Licença	Qtn.	Custo Mensal	Total
IDE VSCode	1	0,00 €	0,00 €
Jira	1	191,67 €	191,67 €
Confluence	1	137,50 €	137,50 €
React License	1	0,00 €	0,00 €
GitLab	1	190,00 €	190,00 €
MUI Pro	1	45,00 €	45,00 €
Total			564,17 €

4.3.2 Materiais

Table 5 - Custos de materiais informáticos

Materiais	Qtn.	Preço Un €	Total
Apple MacBook Air 13.6" M2 CPU 8-core, GPU 8-core SSD 256GB 8GB RAM	10	1 479,00 €	14 790,00 €
Azure Servidor	1	526,12 €	526,12 €
Total			15 316,12 €

5 Planeamento do projeto

5.1 Estrutura Analítica do Trabalho (WBS)

Definiu-se e separou-se o desenvolvimento de todo o projeto no seguinte diagrama WBS:

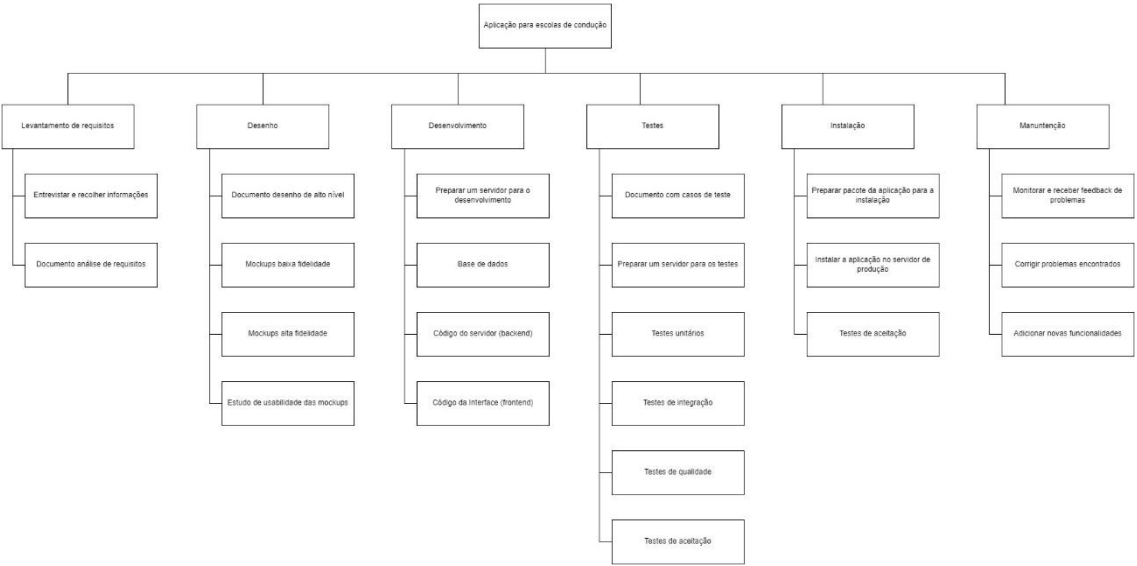


Figure 1 - WBS

5.2 Cronogramas do projeto

5.2.1 Estimativas de trabalho (WBS)

Para os tempos a dedicar a cada tarefa identificada no WBS atribuiu-se um tempo que se acha com a experiência atual que se possui. Abaixo encontra-se uma imagem que é o mesmo gráfico WBS já mostrado, mas possui o tempo previsto em horas representado dentro de um círculo ao lado de cada tarefa.

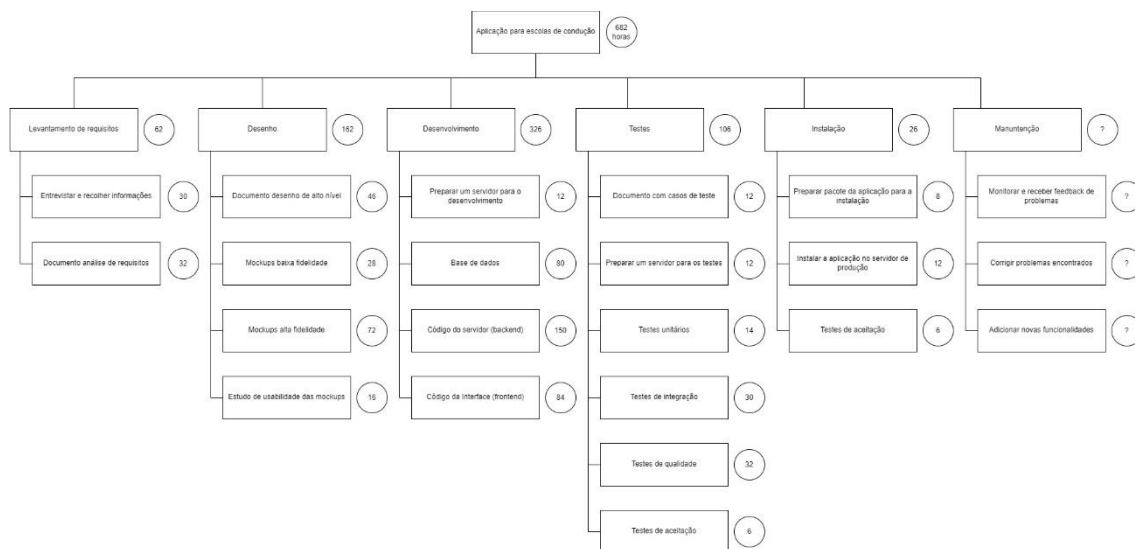


Figure 2 - WBS com estimativa de trabalho

5.2.2 Critical Path Method (CPM)

Na tabela abaixo podemos observar uma representação do CPM em que temos cada tarefa com a sua duração e de que tarefas a mesma depende.

Table 6 - CPM

ID	Tarefa	Tarefas antecedentes (ID)	Duração (horas)
1	Entrevistar e recolher informações		30
2	Documento análise de requisitos	1	32
3	Documento desenho de alto nível	2	46
4	Mockups baixa fidelidade	2,3	28
5	Mockups alta-fidelidade	4	72
6	Estudo de usabilidade das mockups	5	16
7	Preparar um servidor para o desenvolvimento	3	12
8	Base de dados	3,7	80
9	Código do servidor (backend)	3,7,8	150
10	Código da Interface (frontend)	5,6,9	84
11	Documento com casos de teste	9,10	12

ID	Tarefa	Tarefas antecedentes (ID)	Duração (horas)
12	Preparar um servidor para os testes	11	12
13	Testes unitários	12	14
14	Testes de integração	12	30
15	Testes de qualidade	12	32
16	Testes de aceitação	13,14,15	6
17	Preparar pacote da aplicação para a instalação	16	8
18	Instalar a aplicação no servidor de produção	17	12
19	Testes de aceitação fase de produção	18	6

Baseado na tabela acima, com recurso a dois programas gerou-se o grafo do CPM e marcou-se o caminho crítico que nos deu que a duração mínima de projeto, 522 horas, e está representado na imagem abaixo.

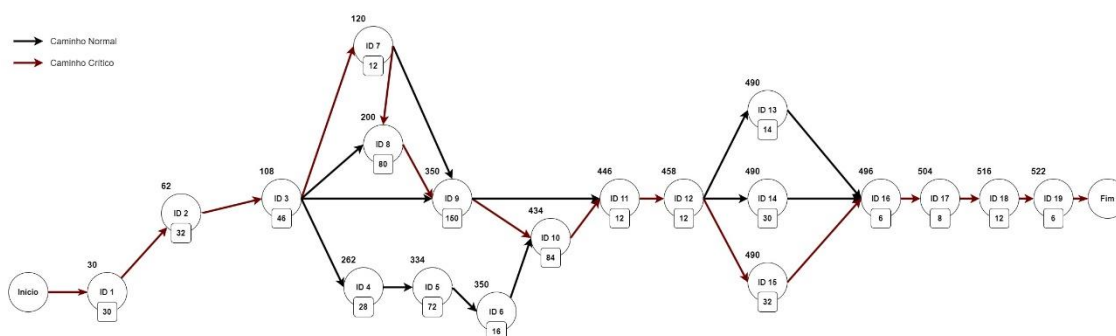


Figure 3 - Caminho Crítico

5.2.3 Diagrama de Gantt

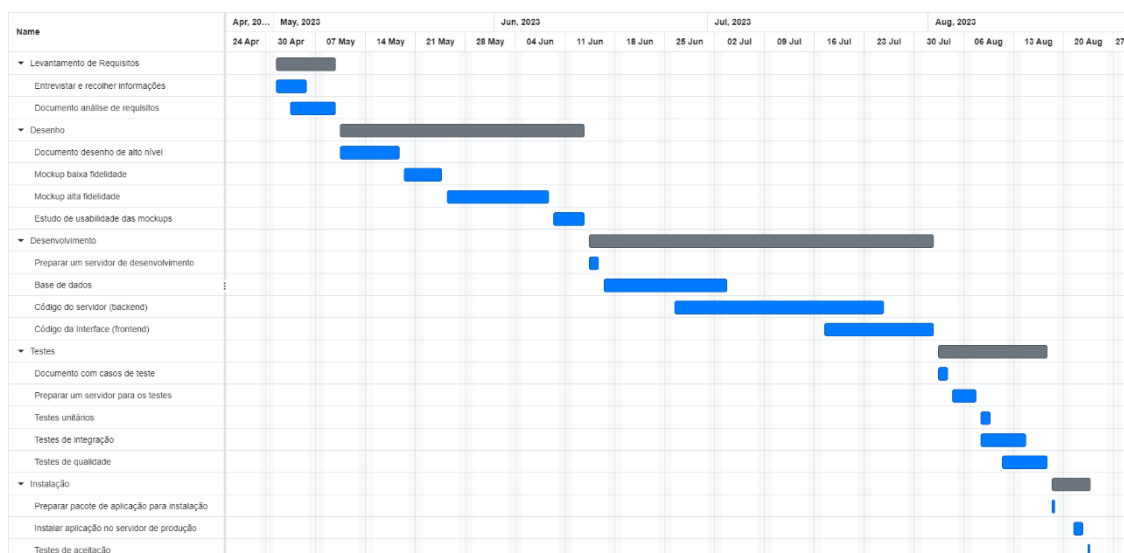


Figure 4 - GANTT

6 Plano de qualidade

A seguir segue-se o plano de qualidade elaborado de forma a assegurar a qualidade do produto final e que está de acordo com os requisitos identificados pelos stakeholders.

1. Padrões de qualidade:

- a. O website deverá ser visualmente atraente e intuitivo.
- b. O website deverá ser acessível.
- c. O website deverá ser responsivo.
- d. O website deverá ter um tempo de resposta inferior a 500 milissegundos.
- e. O website deverá aguentar pelo menos 1000 utilizadores online.
- f. O website deverá estar sempre operacional.
- g. O website deverá conseguir recuperar de uma falha.
- h. O website deverá ser compatível com pelo menos 4 browsers (Chrome, Firefox, Edge e Safari).
- i. O website deverá manter a integridade dos dados.
- j. O website deverá manter a confidencialidade dos dados.
- k. O website deverá ser tolerante aos erros.
- l. O website deverá mostrar ajudas ao utilizador.

2. Riscos potenciais:

Para ver todos os riscos identificados em relação ao projeto ver [Riscos](#).

Aqui são listados alguns riscos relacionados com a qualidade do software.

- a. Complexidade do sistema: Devido ao largo número de funcionalidades, pode existir dificuldades a realizar manter o sistema integrado e livre de erros.
- b. Compatibilidade com diferentes browsers e responsividade: Nem todos os browsers têm a mesma arquitetura nem as mesmas funcionalidades.

Trabalhar com css pode ser complicado tornando a entrega do projeto mais demorada.

3. Plano de qualidade:

- a. **Design Visual:** Serão criadas e testadas mockups de baixa e alta fidelidade que serão desenhadas por um designer profissional. Existirá ainda o acompanhamento de uma equipa de marketing para melhor foco nos clientes finais.
- b. **Acessibilidade:** O website será testado usando ferramentas específicas assegurando que o website é acessível a um maior número de utilizadores.
- c. **Integridade e confidencialidade:** O sistema estará sujeito a testes de força bruta, injeção de dados, entre outros testes para assegurar a segurança do sistema.
- d. **Capacidade:** Serão realizados testes de carga/esforço para assegurar que o sistema é capaz de aguentar grandes volumes de tráfego.
- e. **Testes de aceitação:** O sistema será testado por uma parte da população alvo do sistema.
- f. **Rastreamento de defeitos:** Todos os problemas/defeitos encontrados durante os testes serão rastreados e resolvidos usando ferramentas de rastreamento de defeitos.

4. Controlos de qualidade:

- a. **Ferramentas de versionamento:** Serão utilizadas ferramentas que guardam um histórico das versões.
- b. **Revisão do código:** Antes de um branch ser merged para o main branch será necessário que um conjunto de programadores (i.e. 2) façam a revisão do código, verificando assim se o mesmo está conforme os padrões de qualidade.
- c. **Testes:** O sistema estará sujeito a diversas etapas de testes (unitários, de integração, segurança, carga, sistema e aceitação). Alguns deles serão automatizados.

5. Monitorização e controlo:

- a. **Logs:** O sistema deverá conseguir guardar em logs todos os erros que encontre.

6. Revisão de qualidade final:

- a. Antes do lançamento do produto serão realizados alguns testes e será feita uma revisão para garantir que todos os padrões de qualidade identificados foram cumpridos.

7 Planeamento e gestão de riscos

7.1 Riscos

Table 7 - Riscos

ID	Risco	Ranking
R01	Violação de dados ou acesso não autorizado	1
R02	O website não é responsivo	2
R03	O cliente querer mais funcionalidades do que as originalmente identificadas	1
R04	Paralisia de análise	2
R05	Silos	3
R06	Atraso na entrega do produto	2
R07	Incompatibilidade com certos browsers	3
R08	O sistema apresenta baixo desempenho	2
R09	A equipa ter pouca experiência com as tecnologias a serem utilizadas	3

7.2 Matriz de riscos

Table 8 - Matriz de riscos

Probabilidade	Alta			R01, R03
	Média		R02, R06, R08	R04
	Baixa		R07	R05, R09
		Baixo	Médio	Alto
		Impacto		

Table 9 - O que cada cor da matriz de riscos representa

Cor	Classificação
	1
	2
	3

7.3 Plano de mitigação

Em resposta aos riscos identificados foi criado o seguinte plano de mitigação:

1. R01 - Violação de dados ou acesso não autorizado:
 - a. Usar métodos de segurança já comprovados e sistemas de criptografia, por exemplo 2FA, SSL Protocol e em vez de reinventar a roda usar uma framework comprovada para a autenticação.
 - b. Contratar uma empresa de cyber segurança externa para identificar vulnerabilidades no sistema.
2. R02 - O website não é responsivo:
 - a. Fazer testes à interface.

- b. Contratar programadores de UX/UI experientes.
- 3. R03 - O cliente querer mais funcionalidades do que as originalmente identificadas:
 - a. Deixar em contrato o que vai ser realizado.
 - b. Explicar bem ao cliente o que vai ser feito
 - c. Usar metodologias ágeis para fácil reestruturação das funcionalidades.
- 4. R04 - Paralisia de análise:
 - a. Usar sprints com releases incrementais. Assim não é necessário saber-se tudo logo de início, podendo ir-se moldando o produto.
- 5. R05 – Silos:
 - a. Daily meetings.
 - b. Usar ferramentas de comunicação mais informais, como por exemplo o Discord ou o Whatsapp.
 - c. Incentivar a partilha de ideias e comunicação.
- 6. R06 – Atraso na Entrega do produto:
 - a. Usar cronogramas e marcos
 - b. Usar metodologias ágeis
 - c. Fornecer atualizações regulares sobre o progresso do website com os stakeholders.
- 7. R07 – Incompatibilidade com certos browsers:
 - a. Efetuar testes nos diversos browsers.
 - b. Lançar uma versão beta da aplicação num ambiente de testes e contratar pessoas para a testarem como se a fosse usar no seu dia a dia.
 - c. Receber feedback do utilizador final.
- 8. R08 – O sistema apresenta baixo desempenho:

- a. Realizar testes de carga.
 - b. Monitorizar o desempenho da aplicação e agir sobre esses dados.
 - c. Melhorar a infraestrutura.
9. R09 – A equipa ter pouca experiência com as tecnologias a serem utilizadas:
- a. Contratar programadores experientes nas tecnologias a serem utilizadas.
 - b. Adaptar as tecnologias a serem utilizadas para as quais a equipa se sente mais confortável e ao mesmo tempo não prejudica o desempenho do sistema.
 - c. Formação

8 Conclusão

O grupo acredita que conseguiu atingir as metas propostas.

No entanto, identificou algumas dificuldades na realização do documento, essencialmente devido ao calendário apertado onde se encontrava, não conseguindo tirar dúvidas com o professor.

9 Webgrafia

Documento de apoio dado pelo professor

<https://www.onlinegantt.com/#/gantt>

<https://draw.io>

<https://sanjiverat.com/cpm/>