****

**INSTITUTO POLITÉCNICO DE LEIRIA**

Carbuddy

RELATÓRIO DE projeto em sistemas de informação

Relatório de Projeto em Sistemas de Informação para cumprimento dos requisitos necessários à realização da prova de apresentação de projeto do Curso Técnico Superior Profissional (TeSP) de **Programação de Sistemas de Informação** realizado sob a orientação de Marco Vicente

Andreia Agostinho Batista nº 2202415

diogo Rafael Cunha Pereira nº2201126

gonçalo santos ferreira nº2201131

Programação em Sistemas de informação

DATA

**DECLARAÇÃO**

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

Andreia Agostinho Batista,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Leiria, .... de Fevereiro de 2022

Relatório de Projeto em Sistemas de Informação para cumprimento dos requisitos necessários à realização da prova de apresentação de projeto do Curso Técnico Superior Profissional (TeSP) de **Programação de Sistemas de Informação** realizado sob a orientação de Marco Vicente

**DECLARAÇÃO**

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

Diogo Rafael Cunha Pereira,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Leiria, .... de Fevereiro de 2022

Relatório de Projeto em Sistemas de Informação para cumprimento dos requisitos necessários à realização da prova de apresentação de projeto do Curso Técnico Superior Profissional (TeSP) de **Programação de Sistemas de Informação** realizado sob a orientação de Marco Vicente

**DECLARAÇÃO**

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

Gonçalo Santos Ferreira,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Leiria, .... de Fevereiro de 2022

# Agradecimentos

Para que todo o desenvolvimento deste projeto fosse possível, queríamos deixar um claro agradecimento aos docentes das unidades curriculares do presente ano letivo, sendo eles, Mário Viana, Mário Fernandes e Marco Vicente. Aos restantes elementos envolvidos no projeto, um muito obrigado pelo apoio e ajuda.

Sem os elementos referidos anteriormente, não seria possível desenvolver este projeto com tanta estabilidade e calma. Um muito obrigado a todos eles.

# Resumo

**RELATÓRIO DE PROJETO - CarBuddy**

**Andreia Agostinho Batista**

**Diogo Rafael Cunha Pereira**

**Gonçalo Santos Ferreira**

Em contexto das cadeiras de Plataformas de Sistemas de Informação, Serviços e Interoperabilidade de Sistemas, Acesso Móvel a Sistemas de Informação e Projeto em Sistemas de Informação, foi proposto aos alunos que desenvolvessem um projeto em grupo, projeto esse que irá ser orientado e gerido em concordância entre estas quatro.

O projeto é de tema único que engloba todas e este mesmo é de livre escolha, tal como também se pretende que o desenvolvimento do mesmo utilize uma metodologia ágil.

Todo o projeto será desenvolvido com a separação de tarefas em sprints a serem geridas pelo grupo.

Pode-se referir também o facto de que o projeto tem como principal objetivo a consolidação e aplicação dos conhecimentos adquiridos ao longo do curso.

Por fim, relativamente ao tema do nosso projeto, decidimos implementar um conceito de sistema de gestão e reparação de veículos, que compreende uma combinação de três componentes, sendo eles um website, onde é feita toda a gestão de contas de utilizador, onde vão ser vistos e/ou registados os veículos, o seu estado e as suas reparações, uma aplicação, onde cada cliente, iniciando sessão, tem listadas todas as reparações, dados dos respetivos veículos e o colaborador/mecânico, tal como quando inserindo o VIN de um veículo, acede ao histórico de reparações e aos dados do respetivo veículo. Uma API, para ser possível fazer uma interligação de dados entre ambos os componentes referidos anteriormente.

O objetivo primordial, é que mesmo a nível internacional, se possam registar e verificar o histórico de reparações, utilizando o VIN.

PALAVRAS-CHAVE: gestão, veículos, website, aplicação, API.

# Índices

Índice

[Agradecimentos v](#_Toc93426870)

[Resumo vi](#_Toc93426871)

[Índices viii](#_Toc93426872)

[Formatação do Texto Principal 1](#_Toc93426873)

[1. Introdução 2](#_Toc93426874)

[2. Metodologia 4](#_Toc93426875)

[3. Arquitetura do Sistema 7](#_Toc93426876)

[4. Gestão do Projeto 8](#_Toc93426877)

[5. Análise 9](#_Toc93426878)

[6. Desenho 16](#_Toc93426879)

[7. Implementação 17](#_Toc93426880)

[8. Testes 18](#_Toc93426881)

[9. Conclusão e trabalho futuro 19](#_Toc93426882)

[10. Bibliografia 20](#_Toc93426883)

[11. Anexos 21](#_Toc93426884)

Índice de Ilustrações

[Figura 1 - Divisão de tarefas no Jira 5](#_Toc93426867)

[Figura 2 - Exemplo do Git (Commit's) 6](#_Toc93426868)

[Figura 3 - Gestão temporal do Projeto 6](file:///C:\laragon\www\PSI_PROJETO_ADG\doc\Relatório_FinalPSI_Grupo_ADG.docx#_Toc93426869)

Índice de Tabelas

[Tabela 1 - Requisitos Funcionais 9](#_Toc93426859)

[Tabela 2 - Requisitos não Funcionais de Usabilidade 10](#_Toc93426860)

[Tabela 3 - Requisitos não Funcionais de Fiabilidade 11](#_Toc93426861)

[Tabela 4 - Requisitos não Funcionais de Segurança 12](#_Toc93426862)

[Tabela 5 - Requisitos não Funcionais de Eficiência 12](#_Toc93426863)

[Tabela 6 - Requisitos não Funcionais de Disponibilidade 13](#_Toc93426864)

[Tabela 7 - Requisitos não Funcionais de Ambiente 13](#_Toc93426865)

[Tabela 8 - Requisitos não Funcionais de Desenvolvimento 14](#_Toc93426866)

# Formatação do Texto Principal

Configuração padrão: 1,5 espaço entre linhas; opções de tipo de letra: Garamond, Times New Roman; letra tamanho 12; texto justificado (6 pto antes / 6 pto depois – espaçamento entre parágrafos)[[1]](#footnote-2);

numeração das páginas: árabe, rodapé à direita; letra tamanho 10.

[Esta é sempre a página 1.]

# Introdução

Apresentação do trabalho

Em contexto das cadeiras de Plataformas de Sistemas de Informação, Serviços e Interoperabilidade de Sistemas, Acesso Móvel a Sistemas de Informação e Projeto em Sistemas de Informação, foi proposto aos alunos que desenvolvessem um projeto em grupo, projeto esse que irá ser orientado e gerido em concordância entre estas quatro.

O projeto é de tema único que engloba todas e este mesmo é de livre escolha, tal como também se pretende que o desenvolvimento do mesmo utilize uma metodologia ágil.

Todo o projeto será desenvolvido com a separação de tarefas em sprints a serem geridas pelo grupo.

Pode-se referir também o facto de que o projeto tem como principal objetivo a consolidação e aplicação dos conhecimentos adquiridos ao longo do curso.

Relativamente ao tema do nosso projeto, decidimos implementar um conceito de sistema de gestão e reparação de veículos, que compreende uma combinação de três componentes, sendo eles um website, onde é feita toda a gestão de contas de utilizador, onde vão ser vistos e/ou registados os veículos, o seu estado e as suas reparações, uma aplicação, onde cada cliente, iniciando sessão, tem listadas todas as reparações, dados dos respetivos veículos e o colaborador/mecânico, tal como quando inserindo o VIN de um veículo, acede ao histórico de reparações e aos dados do respetivo veículo. Uma API, para ser possível fazer uma interligação de dados entre ambos os componentes referidos anteriormente.

O objetivo primordial, é que mesmo a nível internacional, se possam registar e verificar o histórico de reparações, utilizando o VIN.

Estrutura do Relatório

Relativamente à estrutura deste documento, podemos observar que este está dividido em 9 pontos, sendo eles a Introdução, a Metodologia, a Arquitetura do Sistema, Gestão do Projeto, Análise, Desenho, Implementação, Testes, e por fim, Conclusão e Trabalho Futuro.

Quanto à Metodologia, podemos referir que falamos a cerca da nossa metodologia de trabalho e de como distribuímos as tarefas entre o grupo.

Na Arquitetura do Sistema trabalhamos os conceitos de componentes tecnológicas utilizadas.

Relativamente à Gestão do Projeto, vamos explicar o nosso método de funcionamento, onde foi adotada uma metodologia ágil, sendo bastante flexível, permitindo contornar eventuais contratempos.

No ponto referente à Análise, tal como está implícito no nome deste mesmo ponto, é feita uma análise pormenorizada dos requisitos e funcionalidades desenvolvidas, bem como dos principais objetivos do projeto.

Apresentamos os Wireframes e Mockups realizados pela equipa no ponto de Desenho, em que também referimos os detalhes e fundamentamos as opções tomadas e as alterações que realizámos para que pudéssemos ao máximo aprimorar o conceito do projeto que escolhemos. Neste ponto também é feita uma comparação entre os desenhos referidos anteriormente, com o resultado final das plataformas desenvolvidas.

Recorremos ao ponto da Implementação para explicar todo o processo de desenvolvimento do software.

No tema dos testes, vamos mostrar todos os testes desenvolvidos ao sistema, de modo a procurar erros e anomalias no código, visando a melhoria do código.

# Metodologia

Durante o período de análise da temática do projeto, o grupo decidiu pensar na melhor forma de abordar o desenvolvimento programático do projeto. Assim, decidimos abordar uma metodologia ágil, que por sua vez é mais flexível.

A metodologia ágil destaca-se pelos seguintes elementos:

* Desenvolvimento incremental e respetiva melhoria contínua,
* Cooperação entre equipa e cliente (ciclo de [feedback](https://neilpatel.com/br/blog/feedback-o-que-e/) constante),
* Entregas rápidas e de alta qualidade,
* Flexibilidade de desenvolvimento do projeto,
* Adaptabilidade às mudanças;

Tendo em conta as vantagens da metodologia ágil discriminadas anteriormente, o grupo destaca um acontecimento real durante o período de desenvolvimento do projeto. Durante essa fase importantíssima do projeto, dois dos elementos do grupo testaram positivo ao covid-19, dificultando o desenvolvimento do projeto, por várias variantes, por causas motoras, físicas e mentais. Felizmente, ambos os casos foram assintomáticos, tornando mais fácil contornar a situação. No entanto, foi importante adotar esta metodologia, pois se molda a situações imprevisíveis.

Ao adotar esta metodologia, o grupo achou, por bem, criar reuniões todos os dias úteis de desenvolvimento, no inicio e no fim do dia. Decidimos também adotar o Git, para a partilhar de código e respetiva documentação e ainda o Jira, para a criação e gestão de todo o tempo útil de desenvolvimento e as respetivas funcionalidades a desenvolver.

Para gerir as tarefas a realizar, utilizámos um repositório do Jira Software. Torna-se mais fácil gerir o projeto, sendo que podemos ver o que falta fazer, o que está em andamento e o que está concluído. Assim fizemos uma divisão de tarefas e distribuímo-las pelos elementos do grupo.

Como podemos ver na seguinte imagem, as tarefas encontram-se listadas e divididas entre todos os membros do grupo no repositório Jira.

Uma imagem com texto

Descrição gerada automaticamente

Figura 1 - Divisão de tarefas no Jira

Como podemos ver na figura abaixo, podemos observar que, todos os membros do grupo fizeram commit’s regularmente, podendo dizer assim, que todos os membros do grupo aproveitaram da melhor forma o seu tempo útil de desenvolvimento.

Foi com esta organização e divisão de tarefas que foi possível desenvolver um projeto com estas três vertentes, sendo elas, API, APP e WEB.

Uma imagem com texto

Descrição gerada automaticamente

Figura 2 - Exemplo do Git (Commit's)

A nível de calendarização estava previsto e foi cumprido, que o grupo durante o mês de outubro analise o problema, faça uma análise de mercado e faça a realização de mockups estruturais do projeto. Em novembro e dezembro o objetivo passa pelo desenvolvimento programático do projeto, complementando-se com correções de erros e bugs e com a escrita do relatório do projeto. Por fim, em janeiro, o objetivo passa pela entrega do respetivo projeto, com toda a sua documentação.



Figura 3 - Gestão temporal do Projeto

# Arquitetura do Sistema

A arquitetura representa as principais componentes tecnológicas que devem ser desenvolvidas no projeto. Deve descrever os componentes técnicos e a forma como interagem entre si.

Modelo relacional e etc…

# Gestão do Projeto

Neste capítulo deve ser apresentado o plano do Projeto (Sugestão: utilizar a ferramenta de Gestão de projetos Redmine):

* Gantt Chart;
* Identificação das principais actividades;
* Deliverables;
* Milestones;
* Equipa de Projeto e funções respetivas.

# Análise

Para proceder ao desenvolvimento do projeto, foi essencial realizar um estudo de requisitos do sistema. Assim, foi realizado um estudo com a finalidade de chegar às funcionalidades essenciais do sistema a desenvolver.

Assim, para esse efeito, foram desenvolvidas tabelas, subdivididas em requisitos funcionais e não funcionais.

Tabela 1 - Requisitos Funcionais

|  |  |  |
| --- | --- | --- |
| # ID | Descrição | Prioridade |
| RF-01 | O sistema deverá obter os seus dados pela API desenvolvida em SIS e pela base de dados | **Alta** |
| RF-02 | O sistema deverá ter Login para todos os utilizadores | **Alta** |
| RF-03 | O sistema deverá ter uma ferramenta de gestão de veículos para os colaboradores | **Alta** |
| RF-04 | O sistema deverá ter uma página onde conste os detalhes de um veiculo selecionado | **Alta** |
| RF-05 | O cliente deverá ter acesso a um activity onde conste os detalhes de um veículo selecionado e as respetivas manutenções e revisões | **Alta** |
| RF-06 | O sistema deverá confirmar todas as ações antes de as executar | **Média** |
| RF-07 | O sistema deverá ter uma resposta de no máximo 2 segundos por cada ação | **Alta** |
| RF-08 | O design do sistema deverá ser simples, agradável e intuitivo | **Alta** |
| RF-09 | O design do sistema deverá ser idêntico ao design do sistema desenvolvido para a unidade curricular de PlatSI | **Média** |
| RF-10 | O sistema deverá estar disponibilizado e terminado em janeiro | **Alta** |
| RF-11 | O sistema deverá estar publicado num repositório GitHub | **Média** |

**Requisitos Não Funcionais de Usabilidade (Usability) –** foca-se na perspetiva da interação com o utilizador (facilidade de uso).

Tabela 2 - Requisitos não Funcionais de Usabilidade

|  |  |  |
| --- | --- | --- |
| # ID | Descrição | Prioridade |
| RNF-USA-01 | **Atividades -** A aplicação deverá ter, no mínimo, 6 atividades e um máximo recomendado de 10 atividades; | **Alta** |
| RNF-USA-02 | **Todas as páginas da aplicação devem ser consistentes -** os elementos devem ser organizados de forma semelhante entre páginas; | **Média** |
| RNF-USA-03 | **Todas as páginas da aplicação devem ser user-friendly –** Devem ser fáceis de usar e apresentar a informação com o número de cliques mínimo; | **Alta** |
| RNF-USA-04 | **O layout das páginas da aplicação deve ser intuitivo –** O layout deve ser apresentado de forma a dar a entender ao utilizador as ações disponíveis através de elementos como botões, mensagens ou erros; | **Média** |
| RNF-USA-05 | **O conteúdo deve ser mostrado de uma forma lógica e natural;** | **Alta** |
| RNF-USA-06 | **Fragmentos -** Algumas funcionalidades têm de ser implementadas/apresentadas através de fragmentos; | **Alta** |
| RNF-USA-07 | **CRUD -** No mínimo, 1 das atividades implementadas deve permitir introdução de dados pelo utilizador; | **Alta** |
| RNF-USA-08 | **Os utilizadores devem conseguir navegar sempre para a homepage** de qualquer parte da aplicação; | **Média** |

**Requisitos Não Funcionais de Fiabilidade (Reliability) –** Descreve-se como a habilidade do software se comportar de forma consistente e aceitável para o utilizador, sendo também a probabilidade e percentagem do software funcionar sem falhas para um período específico ou uso.

Tabela 3 - Requisitos não Funcionais de Fiabilidade

|  |  |  |
| --- | --- | --- |
| # ID | Descrição | Prioridade |
| RNF-FIA-01 | **A aplicação deve ser o mais modular possível**, para que seja fácil a adição de novas funcionalidades; | **Média** |
| RNF-FIA-02 | **A aplicação deve suportar atualizações;** | **Alta** |
| RNF-FIA-03 | **A aplicação deve estar apta a adaptar-se ao aumento do uso** (usage) e a lidar com mais dados à medida que o tempo progride; | **Alta** |
| RNF-FIA-04 | **CRUD -** No mínimo, 4 das funcionalidades implementadas têm de aceder ao webservice para realização das operações CRUD sobre os dados; | **Alta** |
| RNF-FIA-05 | **A aplicação deve ter um tempo de resposta baixo** (até 3 segundos) e todas as ações devem ser processadas instantaneamente; | **Alta** |
| RNF-FIA-06 | **Acesso -** O acesso aos recursos do webservice não pode bloquear a interface gráfica; | **Alta** |

**Requisitos Não Funcionais de Segurança (Security) –** Tais requisitos definem como o sistema se encontra protegido ou como deve ser efetuado o acesso ao sistema.

Tabela 4 - Requisitos não Funcionais de Segurança

|  |  |  |
| --- | --- | --- |
| # ID | Descrição | Prioridade |
| RNF-SEG-01 | **Comunicações -** Tornar todas as comunicações o mais seguras possível; | **Alta** |
| RNF-SEG-02 | **Anti-hacking –** A aplicação deve estar protegida contra ataques exteriores; | **Alta** |
| RNF-SEG-03 | **Cópia de segurança –** A aplicação deve ser capaz de guardar uma cópia local das mensagens e ficheiros enviados e recebidos pelos clientes; | **Alta** |

**Requisitos Não Funcionais de Eficiência (Performance) –** definem o quão bem o sistema lida com capacidade, taxa de transferência e tempo de resposta (desempenho, espaço, tempo).

Tabela 5 - Requisitos não Funcionais de Eficiência

|  |  |  |
| --- | --- | --- |
| # ID | Descrição | Prioridade |
| RNF-EFI-01 | **Taxa de resposta de envio –** Deve conseguir apresentar de imediato as comunicações enviadas pelo cliente; | **Alta** |
| RNF-EFI-02 | **Taxa de resposta de receção –** Deve conseguir apresentar as comunicações recebidas numa fração de segundos; | **Alta** |
| RNF-EFI-03 | **Registo de Dados -** É obrigatório o registo de dados em ficheiros locais e em base de dados SQL local; | **Alta** |
| RNF-EFI-04 | **CRUD -** No mínimo, 1 das atividades implementadas deve permitir introdução de dados pelo utilizador; | **Alta** |

**Requisitos Não Funcionais de Disponibilidade (Availability) –** Estes definem o tempo de funcionamento do sistema, o tempo que leva para reparar uma falha e o tempo entre ciclos.

Tabela 6 - Requisitos não Funcionais de Disponibilidade

|  |  |  |
| --- | --- | --- |
| # ID | Descrição | Prioridade |
| RNF-DIS-01 | **Atualizações -** A aplicação deve estar sempre pronta a obedecer ao seu normal funcionamento, mesmo na ocorrência de atualizações; | **Alta** |
| RNF-DIS-02 | **Erros e Bugs –** De maneira a evitar erros e bugs a equipa deve conseguir editar a aplicação a qualquer momento sem grandes oportunos à utilização normal por parte do utilizador; | **Alta** |

**Requisitos Não Funcionais de Ambiente (Portability) –** especificam o ambiente operacional do sistema (capacidade de adaptação).

Tabela 7 - Requisitos não Funcionais de Ambiente

|  |  |  |
| --- | --- | --- |
| # ID | Descrição | Prioridade |
| RNF-AMB-01 | **Ambiente I -** A aplicação deve correr em sistema operativo Android; | **Alta** |
| RNF-AMB-02 | **Ambiente II -** Uma aplicação Web composta por Front-office e BackOffice; | **Alta** |
| RNF-AMB-03 | **Ambiente III –** A aplicação deve correr em sistemas compatíveis com todos os sistemas operativos; | **Alta** |
| RNF-AMB-04 | **Base de Dados -** Uma base de dados relacional ou NoSQL; | **Alta** |
| RNF-AMB-04 | **API -** Uma API de serviços HTTP; | **Alta** |

**Requisitos Não Funcionais de Desenvolvimento (Implementation) –** indicam restrições ao uso de ferramentas ou linguagens de programação; ou do desenvolvimento do projeto em si; standards.

Tabela 8 - Requisitos não Funcionais de Desenvolvimento

|  |  |  |
| --- | --- | --- |
| # ID | Descrição | Prioridade |
| RNF-DES-01 | **Equipa -** Será valorizada a utilização de planeamento, regras para coordenação entre elementos do grupo e implementação que siga as boas práticas; | **Alta** |
| RNF-DES-02 | **Indentação -** Todos os ficheiros da aplicação devem conter o código devidamente indentado; | **Alta** |
| RNF-DES-03 | **Deadline -** O tempo de desenvolvimento do projeto não deve ultrapassar a data de 24 de Janeiro de 2022; | **Alta** |
| RNF-DES-04 | **Linguagem -** As tecnologias obrigatórias devem consistir em Java, PHP, HMTL; | **Alta** |
| RNF-DES-05 | **Relatório -** No final do projeto deverá ser entregue o respetivo relatório com todos os requisitos desenvolvidos; | **Alta** |
| RNF-DES-06 | **Entrega final I -** No final do projeto deverá ser entregue a aplicação a funcionar devidamente; | **Alta** |
| RNF-DES-07 | **Entrega Final II -** No final do projeto deverá ser entregue o website a funcionar devidamente; | **Alta** |
| RNF-DES-08 | **Entrega Final II -** No final do projeto deverá ser entregue a API a funcionar devidamente. | **Alta** |

Este capítulo deve conter:

* Temática do Projeto
* Objetivos principais
* Análise de requisitos e funcionalidades dos sistemas a desenvolver.

# Desenho

Especificar Protótipos, Casos de Uso, Modelo de Dados, Mockup’s, etc.

# Implementação

Explicar o processo de desenvolvimento do software.

# Testes

Este capítulo deve incluir:

* Desenho de testes
* Ferramentas de testes utilizadas
* Execução dos testes
* Resultados obtidos e correção de erros

# Conclusão e trabalho futuro

A primeira parte de um projeto que está a começar é elaborar um plano para o mesmo. Este plano poderá indicar as diversas etapas que são necessárias passar para que este seja concluído com sucesso. No entanto, existe um elemento que não devemos esquecer no momento de elaborar um plano: todos os dias são diferentes. Isto quer dizer que a motivação não será sempre a mesma, que outras atividades poderão surgir, que podem existir contratempos das mais diversas ordens ou até mesmo os nossos estados emocionais não serão sempre os mesmos todos os dias. Por estes motivos, devemos fazer planos flexíveis e que não causem frustração quando alguma das etapas não for cumprida.

Assim, o nosso tema de projeto não foi escolhido sem ser bem pensado. Pensamos num tema onde todos os elementos do grupo tivessem uma relação positiva e que gostassem desse mesmo tema. Foi fácil manter a motivação, porque, de facto, estávamos a desenvolver algo que queríamos. Por vezes, temos projetos que nos desmotivam devido ao seu tema e não devido ao trabalho e tempo que exigem.

O grupo foi sempre organizado, fluido e dinâmico, o que proporcionou pouco stress no desenvolvimento do projeto e trouxe calmaria no momento do desenvolvimento.

Por fim, é extremamente importante que, no fim do projeto, todos os membros do grupo sintam que deram o seu melhor, e que no fundo, tenham adquirido os conhecimentos relacionados com o projeto. Na atualidade, um programador estende o seu leque de competências com este tipo de projeto, motivando assim o grupo a aprender e dar o seu melhor contributo.

FALAR dA Conclusão e do trabalho future, até agora está so ai a motivação

# Bibliografia

# Anexos

1. Notas de rodapé: fonte 10 pt; 1 espaço entre linhas. [↑](#footnote-ref-2)