Uma imagem com texto

Descrição gerada automaticamente

***Apresentação do Projeto***

**AMSI**



**Nome do grupo: ADG**

Elementos do grupo:

Nº: 2202415 Nome: Andreia Agostinho

Nº: 2201126 Nome: Diogo Pereira

Nº: 2201131 Nome: Gonçalo Ferreira

Torres Vedras, *Janeiro de 2021*

Instituto Politécnico de Leiria

Torres Vedras

Curso Técnico Superior Profissional de Programação de Sistemas de Informação

24 de Janeiro de 2022

Andreia Agostinho

Diogo Pereira

Gonçalo Ferreira

Repositórios



* Repositório Github:

[DiogoRCP/PSI\_PROJETO\_ADG: Projeto final de curso (github.com)](https://github.com/DiogoRCP/PSI_PROJETO_ADG)



* Repositório Jira:

[PSI\_Projeto\_ADG - Quadro ágil - Jira (atlassian.net)](https://psi-da-mds-ga.atlassian.net/jira/software/projects/PPA/boards/4/backlog)

Índice

[Introdução 4](#_Toc92969419)

[1. Contextualização do Projeto 5](#_Toc92969420)

[1.1. Contexto da aplicação 5](#_Toc92969421)

[1.2. Objetivos 6](#_Toc92969422)

[1.3. Motivações 7](#_Toc92969423)

[2. Requisitos Funcionais 7](#_Toc92969424)

[3.1. Opções e Ideias 13](#_Toc92969425)

[3.2. Justificação das Opções/Ideias 14](#_Toc92969426)

[3.3. Requisitos Funcionais Implementados 15](#_Toc92969427)

[3.4. Interface da aplicação 15](#_Toc92969428)

[4. Planeamento e Metodologias 15](#_Toc92969429)

[4.1. Tarefas a realizar e divisão das mesmas 15](#_Toc92969430)

[4.2. Calendarização 17](#_Toc92969431)

[4.3. Tecnologias Utilizadas 18](#_Toc92969432)

[5. Dificuldades e Soluções 18](#_Toc92969433)

[5.1. Apresentação de dificuldades esperadas 18](#_Toc92969434)

[5.2. Propostas de soluções 18](#_Toc92969435)

[5.3. Justificação das Soluções Implementadas 19](#_Toc92969436)

[5.4. Identificação das Funcionalidade propostas, mas não implementadas 19](#_Toc92969437)

[Conclusão 19](#_Toc92969438)

Índice de Tabelas

[Tabela 1 - Requisitos Funcionais 7](#_Toc92969411)

[Tabela 2 - Requisitos não Funcionais de Usabilidade 8](#_Toc92969412)

[Tabela 3 - Requisitos não Funcionais de Fiabilidade 9](#_Toc92969413)

[Tabela 4 - Requisitos não Funcionais de Segurança 10](#_Toc92969414)

[Tabela 5 - Requisitos não Funcionais de Eficiência 10](#_Toc92969415)

[Tabela 6 - Requisitos não Funcionais de Disponibilidade 11](#_Toc92969416)

[Tabela 7 - Requisitos não Funcionais de Ambiente 11](#_Toc92969417)

[Tabela 8 - Requisitos não Funcionais de Desenvolvimento 12](#_Toc92969418)

# Introdução

No âmbito da unidade curricular de Acesso Móvel a Sistemas de Informação do 1º Semestre do 2º ano do Curso Técnico Superior Profissional de Programação de Sistemas de Informação do Instituto Politécnico de Leiria, foi criado o projeto apelidado de “CarBuddy”, que consiste no desenvolvimento de uma plataforma de gestão de todo o tipo de veículos, desde as suas reparações às suas manutenções.

A nossa plataforma consiste numa combinação de três componentes, sendo eles um Website, uma Aplicação Móvel e uma API.

Assim, para a presente unidade curricular, o objetivo é a criação e desenvolvimento da Aplicação Móvel, para a unidade curricular de Serviços e Interoperabilidade de Sistemas, o objetivo é o desenvolvimento da API, sendo esta uma API REST, e para a unidade curricular de Plataformas de Sistemas de Informação, o objetivo é a criação de uma aplicação web.

Na aplicação, é onde cada cliente tem listadas todas as reparações e os respetivos veículos, sendo também possível realizar todas as ações do website na ótica do cliente.

No website, é onde ocorre a gestão de contas de utilizador, onde irão ser vistos/registados os veículos, o seu estado e as suas reparações.

Por fim, para que haja uma interligação de dados entre ambos os componentes referidos anteriormente, procedemos ao desenvolvimento da API REST.

O objetivo, é também que mesmo a nível internacional, se possam registar e verificar o histórico de reparações, centralizando assim todos os dados.

O projeto é comum entre ambas as disciplinas, mas as fases que traçam este projeto são divididas pelas três, conforme a sua pertinência.

Podemos referir também o facto de que o projeto tem como objetivo geral a consolidação de todo o conhecimento adquirido ao longo do curso.

# Contextualização do Projeto

## Contexto da aplicação

O presente relatório é desenvolvido em contexto da cadeira de Acesso Móvel a Sistemas de Informação, e tal como nas restantes cadeiras o projeto é de tema único que engloba todas e este mesmo é de livre escolha. Quanto à presente cadeira, esta terá como papel o desenvolvimento da aplicação móvel do projeto. Com isto, pretende-se também que o desenvolvimento do processo de software utilize uma metodologia ágil. 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. Todo o projeto será desenvolvido com a separação de tarefas em sprints a serem geridas pelo grupo.

Por fim, relativamente ao tema do nosso projeto, decidimos implementar um conceito de gestão de todo o tipo de veículos, ao qual iremos apelidar de CarBuddy.

## Objetivos

O CarBuddy tem como foco a gestão e reparações de qualquer tipo de veículos a nível internacional. Para que tal aconteça, iremos desenvolver uma plataforma segura onde os clientes poderão agendar os serviços nas empresas registadas e poderão gerir todos os seus veículos próprios. Para que isto tenha efeito, pretende-se desenvolver uma aplicação para facilitar a gestão de todo o tipo de veículos e serviços.

Na aplicação o cliente pode agendar um serviço numa empresa registada na plataforma, a sua data e a sua hora. O cliente na aplicação pode adicionar todos os seus veículos, podendo adicionar qualquer tipo de veículos, motas e carros por exemplo, e de qualquer país. Para além disso, pode agendar uma reparação com uma empresa que esteja registada na nossa plataforma. O cliente pode ainda gerir todos os seus veículos, ver os seus dados e o seu histórico de reparações.

É pretendido uma aplicação na qual os clientes, irão fazer o registo dos seguintes dados do utilizador:

* Nome
* Data de Nascimento
* NIF (que deve ser único)
* Password
* E-mail
* Telemóvel.

Estas informações irão ser armazenadas na sua totalidade na base de dados da aplicação gerida pela empresa, de modo a facilitar o pedido de serviços mais do que uma vez.

É de salientar que existem vários tipos de utilizador, administrador, cliente e colaborador, sendo que na aplicação móvel todos os utilizadores são tratados como cliente.

## Motivações

# Requisitos 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 formulário 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 ser alvo de testes de software** | **Média** |
| RF-11 | **O sistema deverá estar disponibilizado e terminado em janeiro** | **Alta** |
| RF-12 | **O sistema deverá estar publicado num repositório GitHub** | **Média** |

1. Requisitos não funcionais

**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** |

## Opções e Ideias

No decorrer do desenvolvimento da aplicação, devem ser especificados o tipo de requisitos e justificar a conformidade com os requisitos obrigatórios estabelecidos na proposta de projeto fornecida pelo docente da cadeira.

Ao analisarmos os objetivos propostos pela proposta de projeto e pelos objetivos práticos que o desenvolvimento da aplicação requer, deparámo-nos com os seguintes requisitos:

* Autenticar-se no servidor fornecendo as credenciais;
* Receber ligações de cliente;
* Guardar a chave pública do cliente;
* Autenticar um utilizador já registado no sistema;
* Validar as assinaturas do cliente;
* Um cliente para dispositivo móvel;
* Uma aplicação Web composta por Front-office e BackOffice;
* Uma base de dados relacional ou NoSQL de suporte às restantes componentes;
* A aplicação deverá ter, no mínimo, 6 atividades (Activity e/ou Fragment) e um máximo recomendado de 10 atividades (Activity e/ou Fragment);
* Algumas funcionalidades têm de ser implementadas/apresentadas através de fragmentos;
* No mínimo, 4 das funcionalidades implementadas têm de aceder ao webservice para realização das operações CRUD (inserir, consultar, alterar e remover) sobre os dados;
* No mínimo, 1 das atividades implementadas deve permitir introdução de dados pelo utilizador;
* No mínimo, 1 das atividades implementadas deve ser construída de forma dinâmica, com os dados obtidos do webservice;
* É obrigatório o registo de dados em ficheiros locais (shared preferences) e em base de dados SQL local (ex.: cache de informação);
* O acesso aos recursos do webservice não pode bloquear a interface gráfica;
* Será valorizada a utilização de planeamento, regras para coordenação entre elementos do grupo e implementação que siga as boas práticas (utilização de padrão MVC, comentários, etc.).

Enquanto grupo, também tivemos algumas ideias, que não são requisitos, mas achamos pertinentes listar, sendo elas:

* Desenvolver a App e o Site com design idênticos,
* Pensar na utilidade da App para o cliente,
* Pensar no futuro e nas valências da App,
* Prever, organizar e dividir as tarefas entre os elementos do grupo,
* Ajudar qualquer um dos membros do grupo, no caso de surgir alguma questão ou dúvida,
* Organizar as ideias de forma sucinta para evitar que surjam conflitos de funcionalidades,
* Realizar pontos de situação todas as semanas,
* Desenvolver o projeto com reuniões semanais,
* Utilização do GitFlow;

## Justificação das Opções/Ideias

As nossas ideias foram registadas logo no início do projeto, como foi referido anteriormente. Fruto de alguma experiência em projetos já realizados no Curso, decidimos implementar e pôr em prática ideias e opções que já tínhamos utilizado.

É importante haver harmonia e consenso no grupo para que tudo corra bem, e todos os assuntos foram sempre debatidos e pensados por todo o grupo.

Assim reforçamos a importância do projeto ser consistente, reforçando a ideia do design da aplicação e do site serem idênticos para o cliente não sentir que está a utilizar duas plataformas diferentes. Decidimos organizar as nossas ideias para ser mais fácil prever as dificuldades que vamos ter. Pensámos no futuro do projeto para chegarmos às funcionalidades que o cliente mais necessita. Realizamos pontos de situação todas as semanas e realizamos reuniões semanais para manter o desenvolvimento integro e organizado. Utilizámos o GitHub para a partilhar de conteúdo entre o grupo e os respetivos professores, mas utilizámos também a tecnologia GitFlow, que permite a criação de várias funcionalidades/branches no Git, tornando tudo mais simples e organizado.

## Requisitos Funcionais Implementados

## Interface da aplicação

# Planeamento e Metodologias

## Tarefas a realizar e divisão das mesmas

Para gerir as tarefas a realizar, utilizámos um repositório do Jira Software. Torna-se mais fácil de 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.

Uma imagem com texto

Descrição gerada automaticamente

A nível de divisão de trabalho, o nosso grupo está a utilizar uma metodologia ágil de desenvolvimento. No entanto o trabalho tem de ser dividido entre todos os 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.

## Calendarização

A nível de calendarização está previsto 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.



## Tecnologias Utilizadas

# Dificuldades e Soluções

## Apresentação de dificuldades esperadas

Relativamente às dificuldades, no decorrer do desenvolvimento da aplicação, tal como as que consideramos que teremos, reunimos os seguintes aspetos:

* Desenvolvimento de layouts: Como não tínhamos conhecimentos para ter como base, houve a necessidade de criação de mockups e layouts do zero, o que demorou algum tempo.
* Desenvolvimento no código: Mesmo com as bases que fomos adquirindo nas aulas, encontrámos algumas barreiras que dificultaram o começo do avanço no código, visto que ainda nos falta aprender mais pontos a cerca do Android Studio.
* Ligação entre páginas (Fragmentos e Activity): Tal como referido acima, a falta de conhecimentos em Android Studio dificultou inicialmente a ligação entre páginas, mesmo tendo em conta que na fase inicial seriam para testar código.
* Organização das Activity/Fragmentos: Saber a ordem das páginas é essencial para podermos organizar a disposição na qual as páginas irão aparecer aquando do lançamento da aplicação.
* Barreira no Desenvolvimento de Diversas Funcionalidades: Por falta de conhecimento.

## Propostas de soluções

Após a análise das dificuldades esperadas, propomos as seguintes soluções:

* Desenvolvimento de layouts: Pesquisa em outras aplicações, vários desenhos e planeamento.
* Desenvolvimento no código: Ajuda prestada por parte do professor da cadeira e procura ativa na internet por meio de comunidades de programação e vídeos.
* Ligação entre páginas (Fragmentos e Activity): Ajuda prestada por parte do professor da cadeira e procura ativa na internet por meio de comunidades de programação e vídeos.
* Organização das Activity/Fragmentos: Entender como funciona o Android Manifest de modo que a organização seja mais intuitiva.
* Barreira no Desenvolvimento de Diversas Funcionalidades: Pesquisa ativa na internet por meio de comunidades de programação e vídeos.

## Justificação das Soluções Implementadas

## Identificação das Funcionalidade propostas, mas não implementadas

# Conclusão