

Apresentação do Projeto

Acesso Móvel a Sistemas de Informação



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

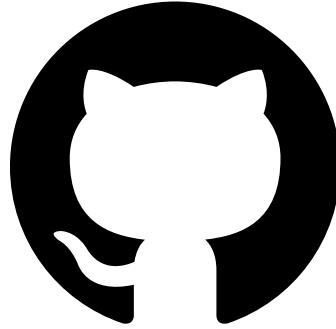
24 de Janeiro de 2022

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://atlassian.net/jira/PSI_Projeto_ADG_-_Quadro_agil)



Índice

INTRODUÇÃO	4
1. CONTEXTUALIZAÇÃO DO PROJETO	5
1.1. CONTEXTO DA APLICAÇÃO	5
1.2. OBJETIVOS	6
1.3. MOTIVAÇÕES.....	7
2. REQUISITOS FUNCIONAIS.....	8
3.1. OPÇÕES E IDEIAS	14
3.2. JUSTIFICAÇÃO DAS OPÇÕES/IDEIAS	15
3.3. REQUISITOS FUNCIONAIS IMPLEMENTADOS	16
3.4. INTERFACE DA APLICAÇÃO	16
4. PLANEAMENTO E METODOLOGIAS	27
4.1. TAREFAS A REALIZAR E DIVISÃO DAS MESMAS	27
4.2. CALENDARIZAÇÃO.....	28
4.3. TECNOLOGIAS UTILIZADAS.....	29
5. DIFICULDADES E SOLUÇÕES	30
5.1. APRESENTAÇÃO DE DIFICULDADES ESPERADAS	30
5.2. PROPOSTAS DE SOLUÇÕES	30
5.3. JUSTIFICAÇÃO DAS SOLUÇÕES IMPLEMENTADAS.....	31
5.4. IDENTIFICAÇÃO DAS FUNCIONALIDADES NÃO IMPLEMENTADAS	31
CONCLUSÃO	32



Índice de Tabelas

Tabela 1 - Requisitos Funcionais	8
Tabela 2 - Requisitos não Funcionais de Usabilidade	9
Tabela 3 - Requisitos não Funcionais de Fiabilidade.....	10
Tabela 4 - Requisitos não Funcionais de Segurança.....	11
Tabela 5 - Requisitos não Funcionais de Eficiência	11
Tabela 6 - Requisitos não Funcionais de Disponibilidade	12
Tabela 7 - Requisitos não Funcionais de Ambiente.....	12
Tabela 8 - Requisitos não Funcionais de Desenvolvimento	13

Índice de Figuras

Figura 1 - Login Activity UI.....	16
Figura 2 - Companies Activity UI.....	17
Figura 3 - SignUp Activity UI	18
Figura 4 - Garage Activity/Fragment UI.....	19
Figura 5 - Vehicle Activity/Fragment UI.....	20
Figura 6 - Repairs Activity/Fragment UI.....	21
Figura 7 - Schedules Appointment Activity/Fragment UI.....	22
Figura 8 - New Vehicle Activity/Fragment UI	23
Figura 9 - Schedules Activity/Fragment UI.....	24
Figura 10 - Account Activity/Fragment UI.....	25
Figura 11 - Menu Principal UI.....	26
Figura 12 - Menu Lateral UI.....	26
Figura 13 - Divisão de tarefas no Jira	27
Figura 14 - Gestão temporal do projeto	28



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.



1. Contextualização do Projeto

1.1. 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.



1.2. 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, indicando a sua data e a sua hora. O cliente na aplicação pode adicionar todos os seus veículos, podendo adicionar de qualquer tipo, tais como motos 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 de 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, sendo eles o administrador, o cliente e o colaborador, sendo que na aplicação móvel todos os utilizadores são considerados clientes.



1.3. Motivações

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 transpor 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 significa 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. 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 calma e ponderação no momento do seu 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 projetos, motivando assim o grupo a aprender e a dar o seu melhor contributo.



2. Requisitos Funcionais

Os requisitos funcionais focam-se no que o sistema deve ou não fazer, facilitando assim o desenvolvimento das funcionalidades.

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 veículo 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 a 24 de Janeiro de 2022	Alta
RF-11	O sistema deverá estar publicado num repositório GitHub	Média



3. 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-05	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 III - No final do projeto deverá ser entregue a API a funcionar devidamente.	Alta



3.1. 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;

3.2. 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 de o 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 íntegro 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.



3.3. Requisitos Funcionais Implementados

A nível de requisitos funcionais foram todos implementados. Felizmente o desenvolvimento programático do grupo começou bastante cedo e foi possível realizar o projeto ao detalhe, com bastante tempo e calma. Assim podemos afirmar que o grupo conseguiu cumprir todos os seus objetivos na realização da aplicação móvel.

3.4. Interface da aplicação

Login Activity2

No login Activity, permitimos ao utilizador realizar a autenticação, através dos seus dados de username e password. A partir desta Activity, que está definida como main no Android Manifest, é ainda possível aceder à atividade de visualização de empresas e à atividade de registo de utilizadores, ambas sem autenticação.

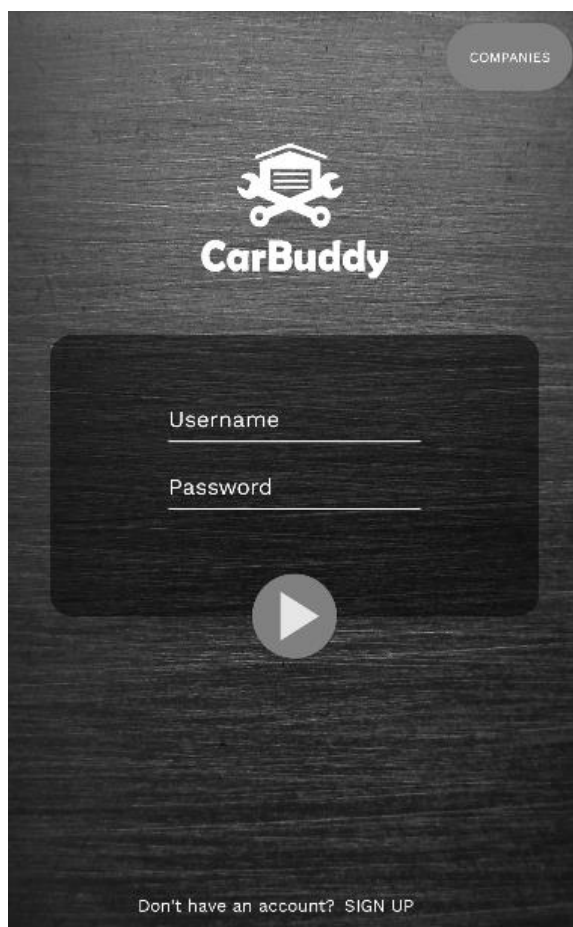


Figura 1 - Login Activity UI



Companies Activity

A partir da atividade de empresas, é possível ao cliente, mesmo sem autenticação, visualizar todas as empresas registadas na plataforma da empresa CarBuddy. Ao utilizador, é ainda possível voltar à atividade de login.



Figura 2 - Companies Activity UI



SignUp Activity

Na atividade de SignUp, o utilizador tem a possibilidade de realizar o seu registo e então, criar a sua conta de cliente. Para esse efeito basta indicar os dados requeridos no formulário e carregar no botão de Signup. Ao utilizador, é ainda possível voltar à atividade de login.



The screenshot shows the SignUp Activity UI. At the top, there is a dark header bar with a back arrow and the text "Signup". Below the header, the form fields are arranged vertically: Username, E-mail, NIF, Phone Number, Password, Repeat Password, and Birthday. The Birthday field is a date picker showing the month, day, and year. At the bottom of the form, there is a large, rounded "Sign Up" button.

Month	Day	Year
Dec	12	2021
Jan	13	2022
Feb	14	2023

Figura 3 - SignUp Activity UI



Garage Activity (Fragment)

Assim que o cliente iniciar sessão, é redirecionado para a atividade/fragmento da garagem, onde estão listados todos os seus veículos. Por cada veículo, serão utilizadas as propriedades marca, modelo, matrícula e tipo de veículo para diferenciar todos os veículos disponíveis na garagem.



Figura 4 - Garage Activity/Fragment UI



Vehicle Activity (Fragment)

Quando o cliente seleccionar um dos seus veículos na garagem, é aberto um novo fragmento com a informação relativa ao veículo seleccionado. A partir desse fragmento é possível ainda ao utilizador aceder aos fragmentos de visualização de reparações do veículo, marcação de agendamentos e edição do mesmo. É ainda possível ao utilizador apagar este veículo, caso o mesmo não tenha reparações e agendamentos.

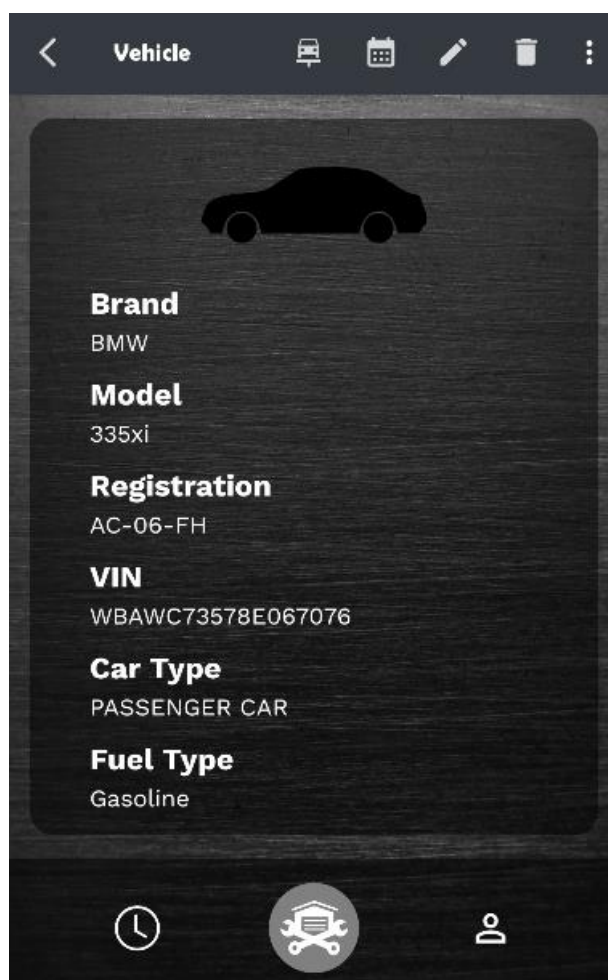


Figura 5 - Vehicle Activity/Fragment UI



Repairs Activity (Fragment)

Quando o utilizador entrar no fragmento de reparações, depois de seleccionar um veículo, poderá ver todas as reparações e manutenções associadas a esse mesmo veículo.

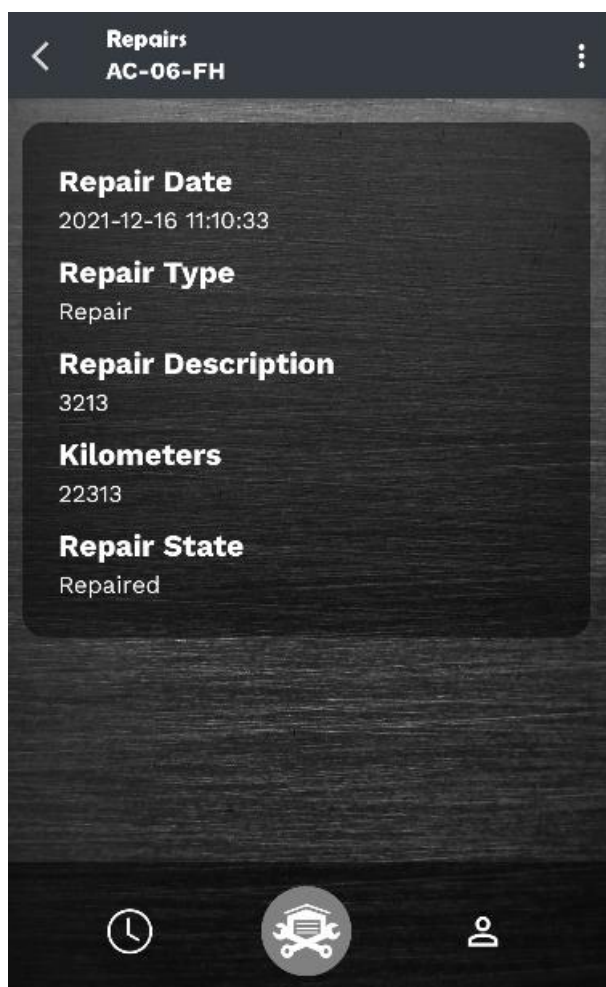
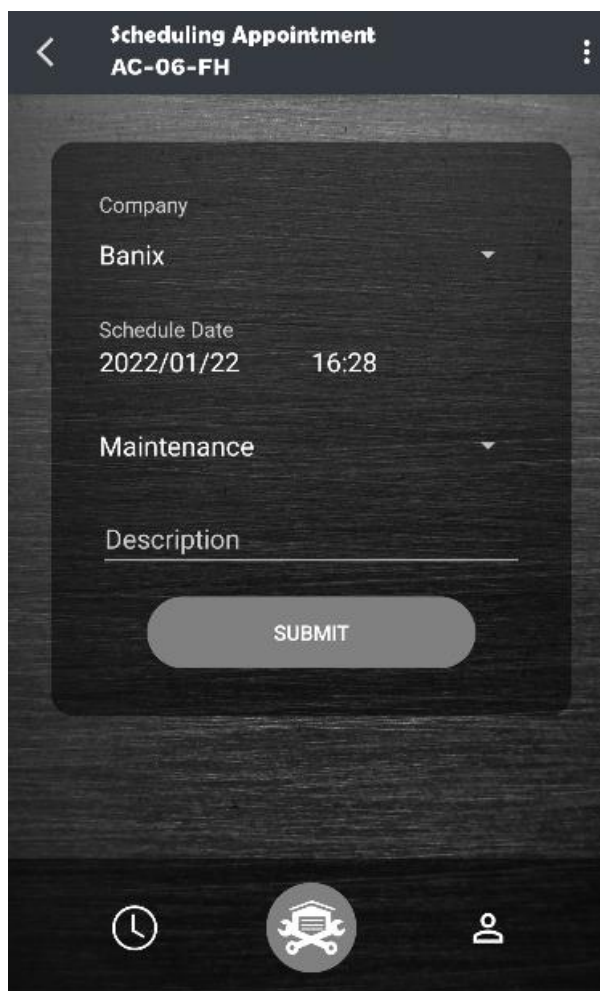


Figura 6 - Repairs Activity/Fragment UI



Schedules Appointment Activity (Fragment)

A partir do fragmento do veículo selecionado, é ainda possível realizar o agendamento de uma reparação/manutenção por parte do cliente. Bastando ao cliente preencher o formulário e submetê-lo.



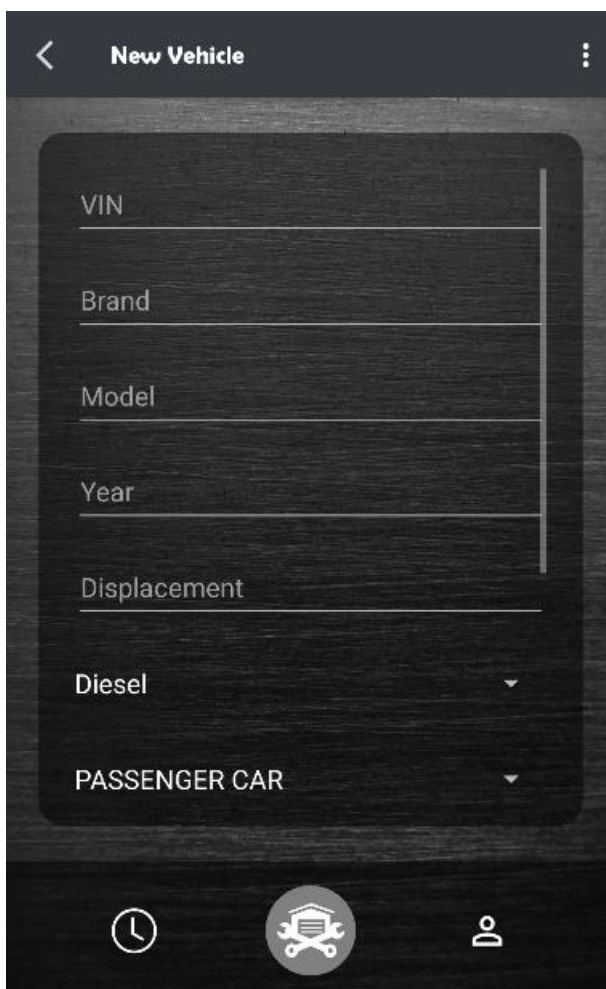
The screenshot displays a mobile application interface for scheduling an appointment. At the top, a dark header bar contains a back arrow, the title "Scheduling Appointment", and the vehicle identifier "AC-06-FH". Below the header, a dark-themed card contains the following fields: "Company" with a dropdown menu showing "Banix", "Schedule Date" with the value "2022/01/22" and a time "16:28", "Maintenance" with a dropdown menu, and a "Description" text input field. A rounded "SUBMIT" button is positioned at the bottom of the card. The bottom of the screen features a navigation bar with three icons: a clock, a car with a wrench (highlighted with a white circle), and a person icon.

Figura 7 - Schedules Appointment Activity/Fragment UI



New Vehicle Activity (Fragment)

Voltando à garagem, o utilizador tem um float action button, que permite adicionar um novo carro. Ao carregar nesse botão, o cliente será redirecionado para o fragmento de adição de um novo veículo. Para adicionar um novo veículo, basta ao cliente preencher os dados do formulário e submetê-lo. É de salientar que este fragmento é utilizado também para a edição de um dado veículo.



The screenshot displays the 'New Vehicle' form within a mobile application. The form is set against a dark, textured background. It features a series of input fields for 'VIN', 'Brand', 'Model', 'Year', and 'Displacement', each with a light-colored underline. Below these are two dropdown menus; the first is currently set to 'Diesel' and the second to 'PASSENGER CAR'. The form is contained within a rounded rectangle. At the top of the screen, a dark header bar contains a back arrow on the left, the title 'New Vehicle' in the center, and a three-dot menu icon on the right. At the bottom, a dark navigation bar contains three circular icons: a clock, a garage with a wrench (the active screen), and a user profile.

Figura 8 - New Vehicle Activity/Fragment UI



Schedules Activity (Fragment)

Todos os fragmentos falados anteriormente, possuem um menu, que permitem uma navegabilidade entre a garagem, entre o fragmento de visualização de todos os agendamentos e o fragmento dos dados da conta. Assim a partir de qualquer fragmento, o cliente pode aceder ao fragmento de visualização de agendamentos, que permite ao cliente ver todos os seus agendamentos.

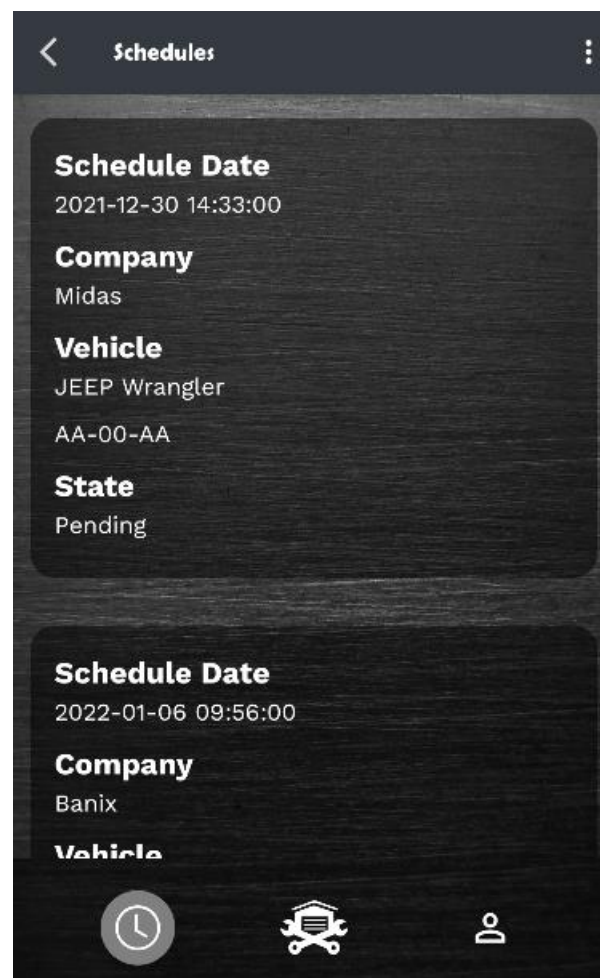


Figura 9 - Schedules Activity/Fragment UI



Account Activity (Fragment)

A partir de qualquer fragmento, o cliente pode aceder ao fragmento de visualização de dados da conta, que permite ao cliente ver todos os seus dados. Permite ainda ao cliente apagar a sua conta caso não tenha veículos associados a ele, e ainda permite a edição dos dados do perfil, utilizando a atividade de registo para esse efeito.

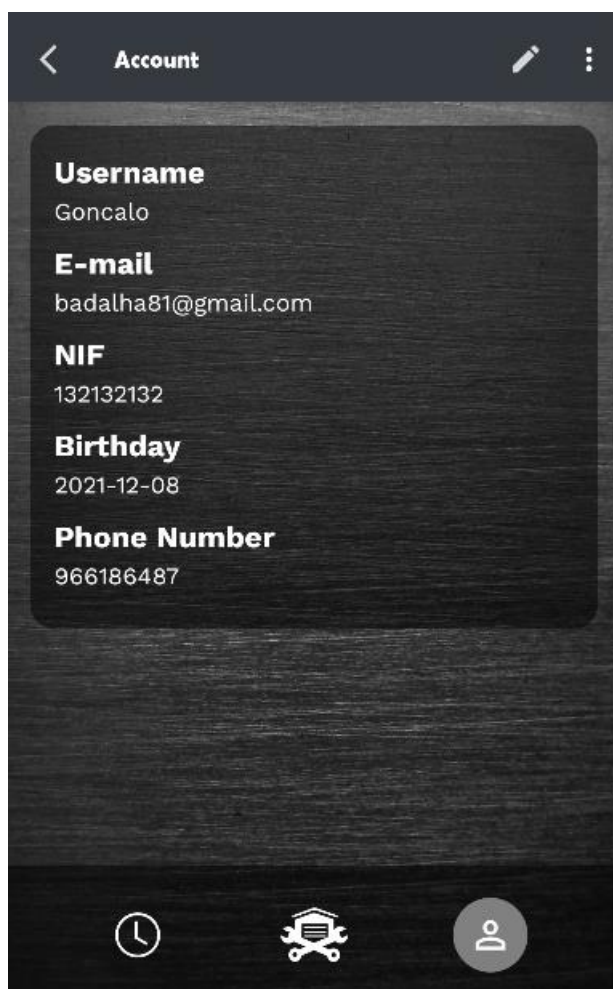


Figura 10 - Account Activity/Fragment UI



Menus

Para facilitar a navegabilidade na aplicação, temos dois menus. O menu principal, tem três botões, que permitem redirecionar o cliente para o fragmento de agendamentos, garagem e conta, respetivamente. É de salientar, que consoante o fragmento ativo, o botão do menu ganha a cor cinzenta.



Figura 11 - Menu Principal UI

A aplicação conta ainda com um menu lateral, onde o cliente pode seleccionar entre a atividade de empresas registadas e, pode também, terminar sessão.

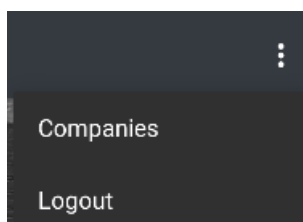


Figura 12 - Menu Lateral UI

É de salientar que toda a aplicação de encontra disponível tanto em idioma português, como em inglês, dependendo do idioma associado ao dispositivo.



4. Planeamento e Metodologias

4.1. 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 distribuímos-las pelos elementos do grupo.

A screenshot of a Jira Sprint board titled 'quadro Sprint 3 19 out - 30 dez (12 problemas)'. The board displays a list of tasks with their status and assigned team members. The tasks are as follows:

Tarefa	Estimativa	Status	Atribuído a
PPA-8 Desenvolvimento Programático do Projeto	30	EM PROGRESSO	[Avatar]
PPA-9 Team Leader (APP)	10	EM PROGRESSO	[Avatar]
PPA-10 Team Leader (API)	10	CONCLUÍDO	[Avatar]
PPA-11 Team Leader (Web)	10	CONCLUÍDO	[Avatar]
PPA-16 Desenvolvimento do Formulário Login	2	CONCLUÍDO	[Avatar]
PPA-17 Desenvolvimento do Formulário de Seleção de Veículos	2	CONCLUÍDO	[Avatar]
PPA-18 Desenvolvimento do Formulário de Informação de Dados do Veículo Selecionado	2	CONCLUÍDO	[Avatar]
PPA-22 Desenvolvimento do Formulário de Registo	2	CONCLUÍDO	[Avatar]
PPA-23 Desenvolvimento do Formulário de Schedules	2	EM PROGRESSO	[Avatar]
PPA-24 Desenvolvimento do Formulário de Visualização de Schedules	2	EM PROGRESSO	[Avatar]
PPA-25 Desenvolvimento do Formulário de Empresas Associadas	2	CONCLUÍDO	[Avatar]
PPA-26 Desenvolvimento do Formulário de Definições	2	CONCLUÍDO	[Avatar]

At the bottom of the board, there is a button labeled '+ Criar problema'.

Figura 13 - Divisão de tarefas no Jira

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.



4.2. 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.

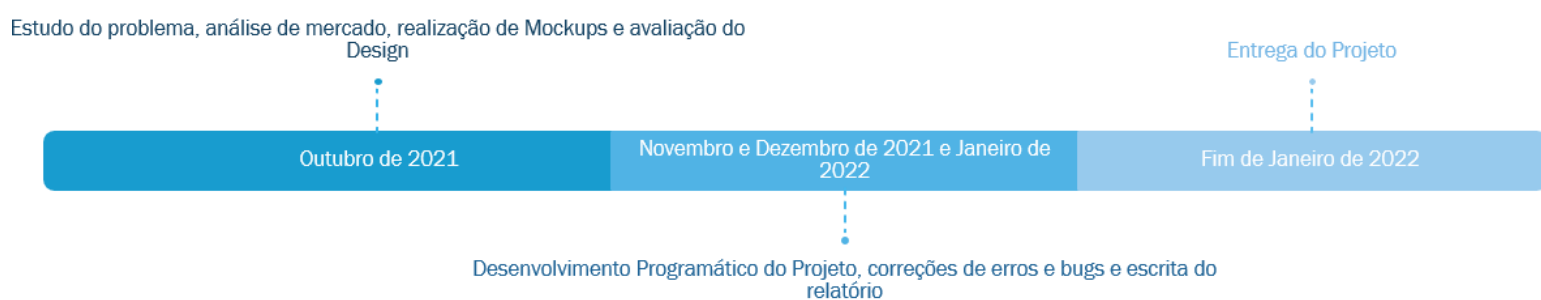


Figura 14 - Gestão temporal do projeto



4.3. Tecnologias Utilizadas

Ao longo da unidade curricular de Acesso Móvel a Sistemas de informação, fomos aprendendo a interagir de uma forma mais profissional com o desenvolvimento em Android.

Assim decidimos utilizar duas tecnologias, sendo elas:

- Singleton - O Singleton é um padrão de desenvolvimento de software. Este padrão garante a existência de apenas uma instância de uma classe, mantendo um ponto global de acesso ao seu objeto. Utilizando o singleton, garantimos a coesão dos dados e a eficiência da aplicação móvel.

- Volley – O Volley é uma biblioteca HTTP que facilita a criação de redes para apps para Android de maneira mais rápida. Utilizando o volley, conseguimos facilmente aceder à API criada para este propósito e garantir a transmissão de dados entre a base de dados da API e o singleton da aplicação móvel.



5. Dificuldades e Soluções

5.1. 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.

5.2. 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.



5.3. Justificação das Soluções Implementadas

As fichas desenvolvidas nas aulas de prática laboratorial foram uma grande ajuda neste ponto. Para além da prática que ganhamos ao desenvolver código, ainda ganhamos um exemplo de código, com boas práticas, para seguir.

Assim, deste modo, decidimos apostar essencialmente na pesquisa ativa na internet quando surgia algum tipo de dúvidas e só depois consultar o professor, no caso da persistência dessa dúvida. Para além da pesquisa ativa na internet, ainda tínhamos as fichas desenvolvidas nas aulas como exemplo e apoio.

De qualquer das formas, conseguimos superar as dificuldades esperadas e ainda aquelas dificuldades de última hora. Infelizmente, por culpa da pandemia, alguns membros do grupo estiveram infetados e ficaram durante algum tempo com menos recursos e possibilidades físicas durante o desenvolvimento, mas nada que no fim não se resolvesse.

5.4. Identificação das Funcionalidades não implementadas

É neste ponto onde estamos verdadeiramente mais motivados. Conseguimos cumprir todos os requisitos iniciais do projeto. No entanto, durante as apresentações da primeira fase, surgiu a ideia de criar agendamentos para possíveis reparações e manutenções, ideia proferida pelo docente da cadeira, que foi adotada de imediato e, por sua vez, desenvolvida.

Para além desta ideia, e tendo em conta um requisito da unidade curricular de Serviços e interoperabilidade de sistemas, foi necessário desenvolver um serviço de notificações utilizando a tecnologia MQTT. Neste caso, na aplicação web, quando o estado de uma reparação ou agendamento é alterado, o cliente, na aplicação, recebe instantaneamente uma notificação com esse propósito.

Por fim, estamos felizes por termos cumprido todos os requisitos do enunciado do projeto, por termos cumprido os requisitos do nosso próprio projeto, e ainda por termos conseguido implementar algumas funcionalidades extra.



Conclusão

Com a elaboração deste projeto, cujo tema seria a criação de uma aplicação móvel onde cada cliente tem listadas todas as reparações e os respetivos veículos, conseguimos desenvolver a prática da programação orientada a objetos dedicada a desenvolvimento de aplicações móveis.

Na primeira fase não tivemos muitas dificuldades na elaboração deste projeto, uma vez que se tratou apenas da análise e criação da sua interface, mas prevíamos grandes desafios na segunda parte do projeto, o que nos fez desenvolver bastante as nossas capacidades na área da programação computacional e na segurança computacional.

Enquanto grupo, o trabalho foi sempre dividido e muito bem gerido entre nós, podendo assim, dizer que ambos os três elementos do grupo trabalharam de igual modo. O grupo fica contente por conseguir concluir assim um projeto com esta grandeza, pois sentimos evolução pessoal e profissional ao realizar este projeto.