|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing icon  Description automatically generated | | Curso de TeSP em  Programação de Sistemas de Informação  PlatSI | |
|  |  | |  |
| **Ano Letivo 2021/2022** | **Av. Periódica** | | **2º Ano, 1º Semestre** |

|  |
| --- |
| **PROJETO DE PlatSI** |

**Relatório de acompanhamento do**

**Ciclo de Desenvolvimento de Software do Projeto em PlatSI**



|  |  |  |
| --- | --- | --- |
| **Turno**: PL1 | **Grupo**: 1 | **Docente**: Mário Fernandes |
| **Nº 2201126** | Diogo Pereira | |
| **Nº 2202811** | Alberto Correia | |
| **Nº 2201131** | Gonçalo Ferreira | |

Instituto Politécnico de Leiria

Torres Vedras

Tesp Programação de Sistemas de Informação

2021-2022

Diogo Pereira

Alberto Correia

Gonçalo Ferreira

**Índice**

[1 Introdução 5](#_Toc85534314)

[1.1 Especificação de Requisitos 6](#_Toc85534315)

[1.1.1 Requisitos Funcionais (RF) 6](#_Toc85534316)

**Índice de tabelas**

[Tabela 1 - Requisitos Funcionais do Back-office da Plataforma Web 6](#_Toc85534318)

[Tabela 2 - Requisitos Funcionais do Front-office da Plataforma Web 8](#_Toc85534319)

# Introdução

No âmbito da unidade curricular de PlatSI do 1º Semestre do 2º ano do Curso TeSP de Programação de Sistemas de Informação do Instituto Politécnico de Leiria, foi criado o projeto “CarBuddy”, que consiste numa plataforma de gestão de todo o tipo de veículos, desde as suas reparações às suas manutenções.

Assim, para a Unidade Curricular de PlatSI, o objetivo do projeto é a criação e desenvolvimento do Website, para a Unidade Curricular de SIS, o objetivo é o desenvolvimento de uma API e para a Unidade Curricular de AMSI, o objetivo é a criação de uma aplicação móvel.

O nosso projeto de gestão de reparação de veículos, consiste numa combinação de três componentes, sendo eles:

Um website, onde se fazem gestão de contas de utilizador, onde vão ser vistos/registados os veículos, o seu estado e as suas reparações.

Uma aplicação, onde cada cliente tem listadas todas as reparações e os respetivos veículos e o colaborador/mecânico, inserindo o VIN do carro acede ao histórico de reparações.

Uma API, para ser possível fazer uma interligação de dados entre ambos os componentes referidos anteriormente.

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

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

## Especificação de Requisitos

### Requisitos Funcionais (RF)

Tabela - Requisitos Funcionais do Back-office da Plataforma Web

|  |  |  |
| --- | --- | --- |
| # 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 estar dividido em back e front office | **Alta** |
| RF-03 | O sistema deverá ter Login para todos os utilizadores | **Alta** |
| RF-04 | O sistema deverá utilizar a tecnologia RBAC para a autenticação | **Alta** |
| RF-05 | O sistema deverá ser desenvolvido utilizando a Framework Yii e a linguagem de programação PHP | **Alta** |
| RF-06 | O back-office deverá estar protegido contra acessos de utilizadores com acesso exclusivo ao front-office | **Alta** |
| RF-07 | O sistema deverá ter uma ferramenta de gestão de utilizadores para o administrador | **Alta** |
| RF-08 | O sistema deverá ter uma ferramenta de gestão de empresas para o administrador | **Alta** |
| RF-09 | O sistema deverá ter uma ferramenta de gestão de veículos para os colaboradores | **Alta** |
| RF-10 | O sistema deverá permitir ao colaborador criar reparações e manutenções | **Alta** |
| RF-11 | O sistema deverá ter uma página onde conste os detalhes de um veiculo selecionado | **Alta** |
| RF-12 | O sistema deverá permitir ao colaborador concluir reparações e manutenções, indicando a descrição das mesmas | **Alta** |
| RF-13 | O sistema deverá confirmar todas ações antes de as executar | **Média** |
| RF-14 | O sistema deverá ter uma resposta de no máximo 2 segundos por cada ação | **Alta** |
| RF-15 | O design do sistema deverá ser simples, agradável e intuitivo | **Alta** |
| RF-16 | O design do sistema deverá ser idêntico ao design do sistema desenvolvido para a unidade curricular de AMSI | **Média** |
| RF-17 | O sistema deverá ser alvo de testes de software (unitários, funcionais e de aceitação) | **Média** |
| RF-18 | O sistema deverá estar disponibilizado e terminado em janeiro | **Alta** |
| RF-19 | O sistema deverá estar publicado num repositório GitHub | **Média** |
| RF-20 | O sistema deverá prevenir hacking SQL Injection | **Baixa** |
| RF-21 | O sistema deverá ser testado utilizando a Framework Codeception | **Média** |
| RF-22 | O sistema deverá encriptar as palavras-passe dos utilizadores | **Alto** |
| RF-23 | O sistema deverá enviar um e-mail ao cliente quando a reparação/manutenção do seu veiculo estiver concluída | **Baixa** |

Tabela - Requisitos Funcionais do Front-office da Plataforma Web

|  |  |  |
| --- | --- | --- |
| # 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 estar dividido em back e front office | **Alta** |
| RF-03 | O sistema deverá ter Login para todos os utilizadores | **Alta** |
| RF-04 | O sistema deverá utilizar a tecnologia RBAC para a autenticação | **Alta** |
| RF-05 | O sistema deverá ser desenvolvido utilizando a Framework Yii e a linguagem de programação PHP | **Alta** |
| RF-06 | O cliente deverá ter acesso a uma página onde conste os detalhes de um veiculo selecionado e as respetivas manutenções e revisões | **Alta** |
| RF-07 | O sistema deverá confirmar todas ações antes de as executar | **Média** |
| RF-08 | O sistema deverá ter uma resposta de no máximo 2 segundos por cada ação | **Alta** |
| RF-09 | O design do sistema deverá ser simples, agradável e intuitivo | **Alta** |
| RF-10 | O design do sistema deverá ser idêntico ao design do sistema desenvolvido para a unidade curricular de AMSI | **Média** |
| RF-11 | O sistema deverá ser alvo de testes de software (unitários, funcionais e de aceitação) | **Média** |
| RF-12 | O sistema deverá estar disponibilizado e terminado em janeiro | **Alta** |
| RF-13 | O sistema deverá estar publicado num repositório GitHub | **Média** |
| RF-14 | O sistema deverá prevenir hacking SQL Injection | **Baixa** |
| RF-15 | O sistema deverá ser testado utilizando a Framework Codeception | **Média** |
| RF-16 | O sistema deverá encriptar as palavras-passe dos utilizadores | **Alto** |