Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática



**Relatório de Base de Dados 2**

Trabalho Prático

Base de Dados 2

Fábio Oliveira Nº 14028

Gabriel Raperger Nº

João Teixeira Nº

Viseu, 2024

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Relatório de Base de Dados 2

Curso de Licenciatura em

Engenharia Informática

Trabalho Prático

Base de Dados 2

Ano Letivo 2023/2024

Fábio Oliveira Nº 14028

Gabriel Raperger Nº

João Teixeira Nº

Viseu, 2024

**Índice**

[1. Introdução 4](#_Toc175322600)

[2. Análise e planeamento 5](#_Toc175322601)

[2.1. Definição do grupo de trabalho 5](#_Toc175322602)

[2.2. Atividades do projeto 5](#_Toc175322603)

[2.3. Gestão das tarefas com Excel 6](#_Toc175322604)

[2.4. Identificação do trabalho individual 6](#_Toc175322605)

[3. Modelo de dados 7](#_Toc175322606)

[3.1. Modelo Físico de dados – Power Designer 7](#_Toc175322607)

[3.2. Diagrama – PgAdmin 7](#_Toc175322608)

[3.3. Diferenças entre os modelos 7](#_Toc175322609)

[4. Desenvolvimento 8](#_Toc175322610)

[4.1. Login 8](#_Toc175322611)

[5. Conclusão 9](#_Toc175322612)

[6. Bibliografia 10](#_Toc175322613)

**Índice de tabelas**

[Tabela 1 - Grupo de trabalho 5](#_Toc175323053)

**Índice de figuras**

[Figura 1‑1. Exemplo de legenda 2](#_Toc56178541)

# Introdução

O presente relatório documenta o desenvolvimento de funcionalidades específicas no âmbito de um sistema de gestão de reparações automóveis, focando-se na implementação de módulos relacionados com o login, gestão de registos e listagem dos mesmos. O trabalho foi desenvolvido com o objetivo de criar uma plataforma robusta e eficiente que permite a gestão de utilizadores, registo de reparações e emissão de faturas, entre outras funcionalidades.

Ao longo do desenvolvimento, foram aplicados diversos conceitos de programação, integrando a utilização da base de dados relacional chamada PostgreSQL e não relacional, MongoDB, bem como a implementação de interfaces de utilizador intuitivas e seguras fazendo uso da Framework Django.

O relatório a seguir detalha o processo de desenvolvimento, desde a concepção e planeamento até a implementação e testes, destacando as soluções adotadas para atender aos requisitos estabelecidos e os resultados alcançados.

# Análise e planeamento

Neste capítulo, abordamos a estruturação do grupo de trabalho, as atividades desenvolvidas e a identificação do trabalho individual. Para garantir uma gestão eficiente e organizada, todas as tarefas foram geridas através do Microsoft Excel, permitindo um acompanhamento detalhado e em tempo real do progresso do projeto.

## Definição do grupo de trabalho

Para o desenvolvimento deste projeto, foi constituído o grupo de trabalho seguinte:

|  |  |
| --- | --- |
| Nome | Número Mecanográfico |
| Fábio Oliveira | 14028 |
| Gabriel Raperger |  |
| João Teixeira |  |

Tabela 1 - Grupo de trabalho

## Atividades do projeto

Ao longo do projeto, foram desenvolvidas diversas atividades, organizadas em fases distintas:

1. **Planeamento Inicial**: Nesta fase, foram definidos os objetivos do projeto, os recursos necessários e o cronograma de atividades em Excel.
2. **Pesquisa e Análise**: Realizou-se uma pesquisa detalhada, seguida de uma análise dos dados recolhidos.
3. **Desenvolvimento e Implementação**: Com base na análise, foram desenvolvidas e implementadas as soluções propostas.
4. **Avaliação e Ajustes**: Após a implementação, as soluções foram avaliadas e ajustadas conforme necessário para garantir a sua eficácia.

## Gestão das tarefas com Excel

Para assegurar uma gestão eficiente das tarefas, utilizámos o Microsoft Excel como ferramenta principal. Através da planilha, foi possível:

* **Atribuir Tarefas**: Cada tarefa foi atribuída a um membro específico do grupo, com prazos e prioridades claramente definidos.
* **Monitorar o Progresso**: Utilizando gráficos e tabelas, acompanhámos o progresso de cada tarefa, identificando possíveis atrasos e ajustando o cronograma conforme necessário.
* **Gerir Recursos**: A planilha permitiu uma gestão eficaz dos recursos.

## Identificação do trabalho individual

Cada membro do grupo de trabalho teve responsabilidades específicas, conforme detalhado abaixo:

COLOCAR IMAGEM DO EXCEL

Esta estrutura permitiu uma clara divisão de responsabilidades, facilitando a gestão do projeto e assegurando que todas as tarefas fossem realizadas de forma eficiente e dentro dos prazos estabelecidos.

# Modelo de dados

Nesta secção, apresentamos o modelo de dados utilizado no projeto, destacando as diferenças entre o modelo inicial, criado no PowerDesigner e o modelo atual, gerado pelo PgAdmin. As alterações no modelo de dados ocorreram devido à migração de tabelas do Django para a nossa base de dados, especialmente pela adoção do sistema de login do Django, o que resultou na inclusão de tabelas adicionais.

## Modelo Físico de dados – Power Designer

COLOCAR IMAGEM DO MFD

## Diagrama – PgAdmin

O diagrama de dados gerado pelo PgAdmin reflete a estrutura atual da nossa base de dados. Este diagrama inclui todas as tabelas e relações que foram adicionadas durante a migração das tabelas do Django. Abaixo, apresentamos o diagrama atualizado:

COLOCAR IMAGEM DO DIAGRAMA

## Diferenças entre os modelos

### Modelo inicial – Power Designer

O modelo inicial, criado no PowerDesigner, foi concebido com base nos requisitos originais do projeto. Este modelo incluía as tabelas e relações necessárias para suportar as funcionalidades básicas do sistema, sem considerar a integração com o Django.

### Modelo atual – PostgreSQL

Com a decisão de utilizar o sistema de login do Django, foi necessário migrar várias tabelas para a nossa base de dados. Esta migração resultou na inclusão de tabelas adicionais, que não estavam presentes no modelo inicial. As principais diferenças incluem:

* **Tabelas de Autenticação**: Tabelas relacionadas ao sistema de login e gestão de utilizadores do Django, como auth\_user, auth\_group, e auth\_permission.
* **Tabelas de Sessão**: Tabelas utilizadas pelo Django para gerir sessões de utilizadores, como django\_session.
* **Outras Tabelas**: Tabelas adicionais que suportam funcionalidades específicas do Django, como django\_migrations.

A migração das tabelas do Django para a nossa base de dados foi uma decisão estratégica para aproveitar as funcionalidades robustas de autenticação e gestão de utilizadores oferecidas pelo Django. Embora isso tenha resultado em um modelo de dados mais complexo, acreditamos que os benefícios superam as desvantagens, proporcionando uma base de dados mais segura e eficiente.

# Desenvolvimento

Nesta secção, vamos detalhar cada funcionalidade desenvolvida. As imagens da aplicação serão inseridas após a explicação de cada funcionalidade, para facilitar a compreensão.

## Login

O sistema de login foi desenvolvido para garantir a segurança e a integridade dos dados dos utilizadores. Este módulo permite que utilizadores, divididos em diferentes grupos acedam às funcionalidades da plataforma de acordo com as suas permissões. Os grupos são:

* Administrador
* Cliente
* Trabalhador

O sistema foi concebido para gerir as credenciais dos utilizadores através da tabela *auth\_user*, integrada na base de dados PostgreSQL. Além disso, foi implementado um script para migração de dados e criação automática dos grupos de utilizadores, facilitando a gestão de permissões e simplificando a atribuição de grupos aos novos utilizadores.

A criação de grupos no sistema visa a organização dos utilizadores, permitindo que cada grupo tenha acesso exclusivo a determinadas funcionalidades. Por exemplo, os administradores têm acesso total à plataforma, enquanto os encarregados e clientes têm acessos limitados às funcionalidades relacionadas com as suas responsabilidades e necessidades.

Para exemplificar, foi criado um script que, ao ser executado, cria automaticamente os grupos necessários e atribui as permissões adequadas a cada grupo. Este processo de configuração inicial é fundamental para a correta operação do sistema.

**Mostrar a página de login**

## Clientes

## Encarregados

# Testes

# Conclusão

O desenvolvimento das funcionalidades descritas neste relatório contribuiu significativamente para a criação de uma plataforma eficiente e robusta para a gestão de reparações automóveis. As funcionalidades de login, gestão de clientes e encarregados feitas com recurso a Django e a integração com bases de dados PostgreSQL e MongoDB, demonstraram ser essenciais para o bom funcionamento do sistema.

Durante o desenvolvimento, foram enfrentados desafios técnicos relacionados com a integridade referencial, a gestão de permissões de utilizadores, e a integração entre diferentes tecnologias de bases de dados. No entanto, através de uma abordagem estruturada e detalhada, foi possível superar esses desafios, resultando num sistema funcional e preparado para ser utilizado na prática.

A realização dos testes aos objetos lógicos confirmou a robustez das implementações e assegurou que todas as funcionalidades operam conforme esperado. O resultado final é uma aplicação que não só atende às necessidades dos utilizadores, como também está preparada para futuras expansões e melhorias.

# Bibliografia