Projeto Temático em Desenvolvimento de Aplicações

Relatório Final

**Gestão de Tráfego - Radares**

|  |  |
| --- | --- |
| Grupo: | Austin Foote nº84935  Bruno Ferreira nº95291  Carolina Tavares nº92658  Diogo Oliveira nº95182  Maria Nobre nº92659 |
|  |  |
| Versão: | v |
| Orientador: | Joaquim Ferreira  Ano letivo 2019/2020 |

Índice

[1 Introdução 6](#_Toc29060375)

[1.1 Visão geral do sistema 7](#_Toc29060376)

[1.2 Cliente 8](#_Toc29060377)

[1.3 Objetivos 8](#_Toc29060378)

[2 Planeamento 9](#_Toc29060379)

[3 Modelo de requisitos 11](#_Toc29060380)

[3.1 Requisitos funcionais 11](#_Toc29060381)

[2.2 Restrições e requisitos não funcionais 12](#_Toc29060382)

[3.1.1 Requisitos de interface e facilidade de uso 12](#_Toc29060383)

[3.1.2 Requisitos de desempenho 13](#_Toc29060384)

[3.1.3 Requisitos de segurança e integridade dos dados 13](#_Toc29060385)

[4 Modelo de Casos de Utilização 14](#_Toc29060386)

[4.1 Visão geral 14](#_Toc29060387)

[4.2 Atores 14](#_Toc29060388)

[4.3 Descrição dos casos de utilização 15](#_Toc29060389)

[4.3.1 [Estatísticas de Tráfego #1] 15](#_Toc29060390)

[4.3.2 [Configuração Sistema #2] 16](#_Toc29060391)

[4.3.3 [Consultar informações do tráfego #3] 17](#_Toc29060392)

[4.3.4 [Autenticação #4] 17](#_Toc29060393)

[4.4 Cobertura de requisitos 18](#_Toc29060394)

[5 Diagrama de Classes 19](#_Toc29060395)

[6 Diagrama de Entidades 20](#_Toc29060396)

[7 Implementação da Base de Dados 21](#_Toc29060397)

[7.1 Criação das tabelas 21](#_Toc29060398)

[7.2 Criação das views 21](#_Toc29060399)

[7.3 Filtração e inserção dos dados do radar na base de dados 21](#_Toc29060400)

[8 Principais Métodos a Implementar 22](#_Toc29060401)

[9 Testes Unitários 24](#_Toc29060402)

[10 Implementação da aplicação no netbeans 25](#_Toc29060403)

[10.1 Aplicação Cliente – Servidor 25](#_Toc29060404)

[10.2 Conexão à base de dados – JDBC 25](#_Toc29060405)

[10.3 Conexão à base de dados – JDBC 25](#_Toc29060406)

[10.4 Criação interfaces em JFrame 25](#_Toc29060407)

[10.5 Implementação classes e métodos 25](#_Toc29060408)

[10.5.1 Mostrar estatísticas simples 25](#_Toc29060409)

[10.5.2 Mostrar histórico 25](#_Toc29060410)

[10.5.3 Autenticação 25](#_Toc29060411)

[10.5.4 Registo de Entidades 25](#_Toc29060412)

[11 Testes?? 26](#_Toc29060413)

[12 Análise dos Resultados 27](#_Toc29060414)

[13 Reflexão Crítica e Conclusão 28](#_Toc29060415)

[14 Anexos 30](#_Toc29060416)

Índice de Figuras

[Figura 1 - Cronograma 9](file:///C:\Users\Maria%20Nobre\Documents\Git\Maquinas\Relatório\RelatórioProjeto3.0.docx#_Toc29060417)

[Figura 2 - Diagrama Casos de Utilização 14](file:///C:\Users\Maria%20Nobre\Documents\Git\Maquinas\Relatório\RelatórioProjeto3.0.docx#_Toc29060418)

[Figura 3 - Diagrama de Atividades Estatísticas de Tráfego 15](file:///C:\Users\Maria%20Nobre\Documents\Git\Maquinas\Relatório\RelatórioProjeto3.0.docx#_Toc29060419)

[Figura 4 - Diagrama de Atividades Configuração Sistema 16](file:///C:\Users\Maria%20Nobre\Documents\Git\Maquinas\Relatório\RelatórioProjeto3.0.docx#_Toc29060420)

[Figura 5 - Diagrama de Classes 19](file:///C:\Users\Maria%20Nobre\Documents\Git\Maquinas\Relatório\RelatórioProjeto3.0.docx#_Toc29060421)

[Figura 6 - Diagrama de entidades 20](file:///C:\Users\Maria%20Nobre\Documents\Git\Maquinas\Relatório\RelatórioProjeto3.0.docx#_Toc29060422)

Índice de Tabelas

[Tabela 1 - Requisitos Funcionais 8](#_Toc23938949)

[Tabela 2 - Caso de Utilização #1 – Estatísticas de Tráfego 12](#_Toc23938950)

[Tabela 3 - Caso de Utilização #2 - Configuração Sistema 13](#_Toc23938951)

[Tabela 4 - Caso de Utilização #3 - Consultar Informações do Tráfego 14](#_Toc23938952)

[Tabela 5 - Caso de Utilização #4 - Autenticação 14](#_Toc23938953)

# Introdução

No âmbito do Projeto Temático em Desenvolvimento de Aplicações, foi proposto ao grupo conceber uma aplicação, cujo tema escolhido foi “Gestão de Tráfego por Radares”. Este tema consiste na utilização dos dados fornecidos por radares para gerar estatísticas relevantes aos condutores e outras organizações interessadas em estudar o tráfego na zona.

Os dados serão fornecidos ao grupo pelo gestor dos radares da Barra e Costa Nova, que, por sua vez, são colocados numa base de dados criada pelo grupo. A aplicação irá utilizar estes dados para gerar e mostrar estatísticas, tais como velocidade média e contagem de veículos. Com estes dados, os utilizadores da app poderão determinar se existe trânsito na zona ou não. Estas estatísticas também poderão ser partilhadas com mais detalhe para organizações, através de uma interface de login, sendo que as credenciais são fornecidas pelo administrador do sistema.

Estando este projeto incluído no módulo Temático em Desenvolvimento de Aplicações, juntamente com as disciplinas de Engenharia de Software e de Sistemas de Bases de Dados, tinha-se também como objetivo, aplicar os conhecimentos adquiridos nestas aulas.

Ao longo deste relatório será, então, descrito todo o processo de desenvolvimento deste projeto, desde o seu planeamento, definição dos requisitos, planificação e divisão de tarefas, até à implementação da aplicação.

Para finalizar, será feita uma análise dos resultados, uma pequena reflexão crítica e conclusão.

## Visão geral do sistema

Foram colocados vários radares, na Barra e Costa Nova, estes radares detetam os objetos que estão a passar na zona. Identificam os objetos distinguindo se se trata de um carro, de um pesado, de uma bicicleta/mota ou de um peão. Outros objetos ficam na categoria de objetos não identificados, por exemplo, animais, árvores, etc. O radar consegue acompanhar estes objetos durante um certo troço da estrada, fazendo mais que uma medição nesse troço, sendo que quanto mais tempo acompanhar o objeto mais correta será a medição. Para além da sua categoria, de cada objeto são guardadas mais informações, nomeadamente, a data, hora e zona (do radar) da medição, o sentido e a que velocidade vai. Assim, sabendo quantos carros é que estão a entrar naquela zona e sabendo se estão ou não a sair, percebesse se existe congestionamento na zona.

Os radares após captarem estes dados guardam-nos num ficheiro de texto. O grupo vai trabalhar com os dados a partir destes ficheiros, com o objetivo de criar uma aplicação que informe os seus utilizadores destas várias estatísticas. O grupo configurará este sistema utilizando comunicação por sockets, modelo cliente-servidor. O sistema tem de ser configurado de modo a saber quantos radares existem, onde é que se encontram e quantos sentidos e vias existem na estrada. Os dados dos ficheiros provenientes do radar são colocados numa base de dados própria, que depois será usada pela aplicação, esta tem de ser configurada de modo a fazer os cálculos necessários para mostrar as várias informações num dado intervalo de tempo.

Relativamente à aplicação esta terá dois tipos de utilizadores, os com e os sem registo, assim, tem de se criar uma lista dos utilizadores com os seus respetivos logins e passwords. As informações que cada um dos tipos de utilizadores terão acesso não serão as mesmas, pois as suas necessidades também serão diferentes.

O utilizador sem registo será, por exemplo, um condutor normal que quer ir passear e procura saber se ir para a praia será uma boa ideia naquela hora ou se estará muito trânsito. Assim, ele entra na aplicação rapidamente (sem precisar de fazer login), procura a informação que necessita, que neste caso, é a contagem dos carros e a velocidade média no momento. Sabendo isto, ele depois toma a decisão de ir ou não para a praia.

O utilizador com registo, por exemplo o estado, pretende perceber se aquela zona necessita de melhorias no que toca ao controlo do congestionamento. Eles irão entrar na aplicação, usando credenciais fornecidas pelo administrador, após terem-se registado previamente. De seguida, eles irão consultar o histórico das velocidades. Para perceberem quais as horas e dias em que houve mais movimento e qual a quantidade de carros que circulam na zona. Assim sabem se será necessário tomar certas medidas para melhorar a circulação nestas vias, melhorar a forma de entrada na zona, ou então irão usar estes dados apenas para efeitos estatísticos.

## Cliente

Para este projeto, o cliente é a empresa que contratou o grupo para criar esta aplicação. Esta empresa pretendia que fosse criada uma aplicação que fizesse a gestão do tráfego nas diversas zonas onde estão colocados os seus radares. Deixando a critério do grupo quais as estatísticas a serem geradas, quais os seus requisitos e casos de utilização. O cliente não necessita de ter acesso a como são geradas as estatísticas, apenas precisa de configurar a aplicação para funcionar para cada um dos diferentes radares.

## Objetivos

Assim, tendo em conta a visão geral deste projeto podem-se agrupar os seguintes objetivos:

* Conceber uma aplicação que permita a gestão do tráfego para qualquer zona à escolha do administrador;
* Permitir a qualquer condutor ter a possibilidade de tomar decisões com base na informação do trânsito obtida pela aplicação;
* Permitir a certas entidades o acesso a informações mais detalhadas do trânsito na zona, para estudos estatísticos ou, eventualmente, para melhorar a circulação do trânsito na área.

# Planeamento

Relativamente ao planeamento do projeto, optou-se por seguir um método ágil. Sendo assim, o planeamento foi dividido em várias iterações.

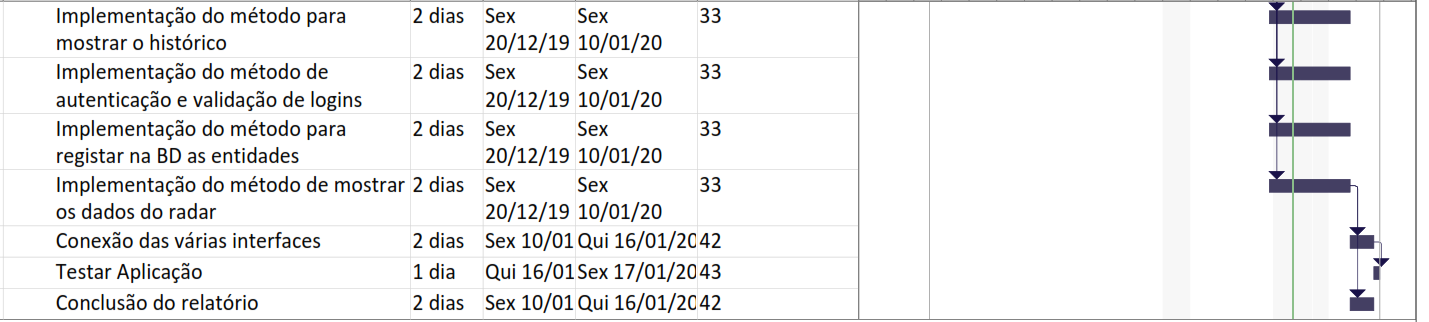
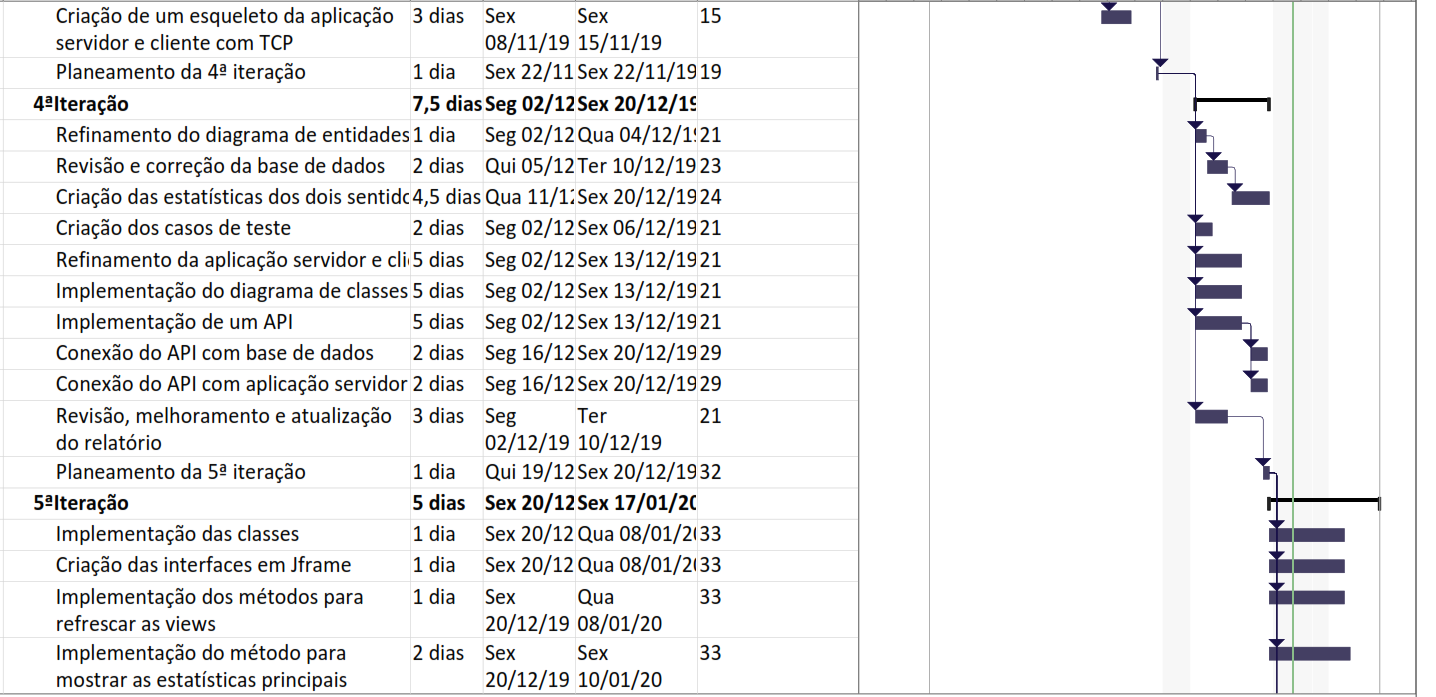
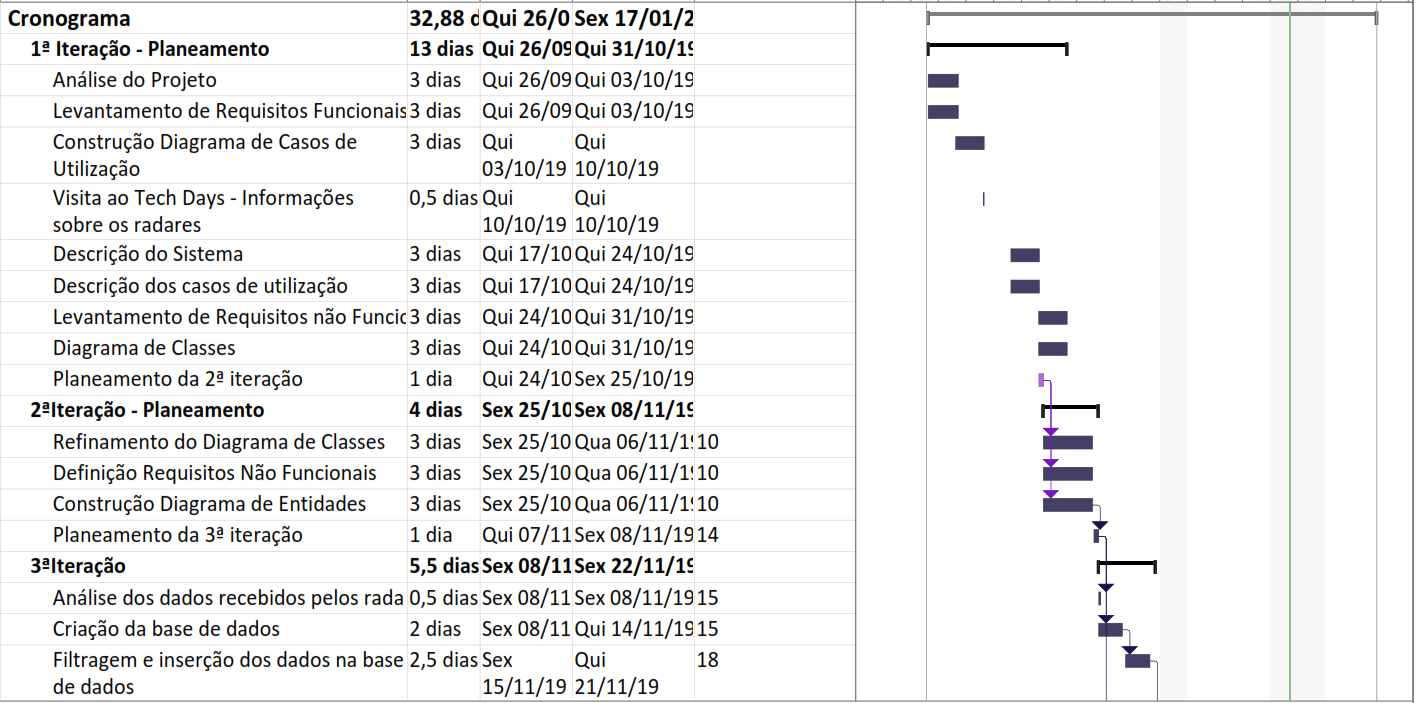


Figura 1 - Cronograma

Para cada uma das iterações, iam sendo definidas várias tarefas e respetivas durações estimadas.

Nas primeiras fases, durante o planeamento, definiram-se os objetivos do trabalho, fazendo-se o levantamento dos casos de utilização e dos requisitos da aplicação. Tendo isto foram elaborados vários diagramas. Começando-se pelo diagrama de casos de utilização, para uma melhor descrição de algum dos casos foram feitos diagramas de atividades. Posto isto, elaborou-se o diagrama de classes, com os seus respetivos atributos e métodos mais relevantes. De seguida, elaborou-se o diagrama de entidades, o diagrama físico da base de dados. O planeamento do projeto levou mais tempo do que devia, o que atrasou um pouco o resto do projeto.

Usando um método ágil, a implementação do projeto deve ser dirigida por testes, assim feito o planeamento passar-se-ia à criação dos casos de teste e de seguida à implementação e programação das várias classes, métodos, JFrame e criação da base de dados.

# Modelo de requisitos

## Requisitos funcionais

A seguinte tabela trata-se de um inventário das funções que o sistema tem de suportar.

Tabela 1 - Requisitos Funcionais

|  |  |  |
| --- | --- | --- |
| Refª | Requisito funcional | Prioridade |
| RF.1 | Permite a consulta das informações relativas às velocidades das viaturas em ambos os sentidos (histórico de velocidades). | +++ |
| RF.2 | Permite a consulta da contagem das viaturas que passam pelo radar (tráfego). | +++ |
| RF.3 | Permite a consulta da velocidade média em intervalos de 10 em 10 minutos. | +++ |
| RF.4 | Sistema indica a velocidade máxima e mínima. | ++ |
| RF.5 | Os condutores (sem registo) só terão acesso à velocidade média de trânsito (de 10 em 10 min) e ao contador de viaturas. | +++ |
| RF.6 | As entidades registadas (Governo, IMT, Brisa…), terão acesso a todas as estatísticas disponíveis. | +++ |
| RF.7 | Permitir a configuração da aplicação, de modo a funcionar para qualquer radar, independentemente da sua localização. | - |
| RF.8 | Permitir o registo e eliminação de entidades. | ++ |

## 2.2 Restrições e requisitos não funcionais

### Requisitos de interface e facilidade de uso

Requisitos que não são essenciais para o funcionamento do serviço, mas que facilitam a navegação da interface para o utilizador

|  |  |
| --- | --- |
| Refª | Requisito de interface e usabilidade |
| RInt.1 | Dados têm de se atualizar de 10 em 10 minutos. |
| RInt.2 | A interface do condutor é a interface principal, sendo logo mostrada quando se abre a aplicação. |
| RInt.3 | A interface do condutor possui um mapa com a localização do radar, tendo também indicação do número de vias e sentidos nessa zona. |
| RInt.4 | Mapa tem de ser manipulável, mas será possível afastar-se muito da zona do radar. |
| RInt.5 | A interface do condutor possui também uma tabela com as informações da velocidade média para cada um dos sentidos, a informação da contagem dos carros, mostrando apenas a diferença e a avaliação do trânsito (muito trânsito, trânsito moderado e pouco trânsito). |
| RInt.6 | A interface das entidades possui uma tabela com as velocidades máxima e mínima e um gráfico com as velocidades médias de 10 em 10 minutos. Pode também ser aberto noutra página uma tabela com um histórico de todas as velocidades. |
| RInt.7 | O administrador tem ainda acesso a uma interface de configuração onde faz o registo das novas entidades. |

### Requisitos de desempenho

Requisitos e restrições que não são essenciais para o serviço, mas que diminuem a carga de processamento do servidor, evitando situações que atrasam a resposta de servidor

|  |  |
| --- | --- |
| Refª | Requisito de desempenho |
| RDes.1 | Limite máximo de 254 clientes ligados ao mesmo tempo |
| RDes.2 | Existir um servidor adicional caso o primário ir abaixo |

### Requisitos de segurança e integridade dos dados

[relacionar requisitos de segurança com os requisitos funcionais identificados, quando aplicável]

|  |  |
| --- | --- |
| Refª | Requisito de segurança, privacidade e integridade de dados |
| RSeg.1 | Não são guardados qualquer tipo de dados dos condutores. |
| RSeg.2 | Os dados das entidades são privados e não podem ser partilhados. |

# Modelo de Casos de Utilização

## Visão geral

Figura 2 - Diagrama Casos de Utilização

## Atores

| Ator | Descrição |
| --- | --- |
| **Radar** | Equipamento que recolhe dados brutos do tráfego na zona. |
| **Administrador** | Indivíduo que configura o sistema e faz a sua manutenção. |
| **Condutores** | Utilizadores normais que pretendem fazer consultas simples de tráfego. |
| **Entidades Registadas** | Utilizadores com login que têm acesso a informações mais detalhadas. |

## Descrição dos casos de utilização

### [Estatísticas de Tráfego #1]

Tabela 2 - Caso de Utilização #1 – Estatísticas de Tráfego

| Nome: |  |
| --- | --- |
| Atores: | Radar (inicia) |
| Prioridade: | Alta |
| Finalidade: | Processar os dados do radar para gerar estatísticas de tráfego. |
| Requisitos funcionais: |  |
| Pré-condições: | Receber dados provenientes do Radar. |
| Sumário: | Os dados fornecidos pelo radar serão processados pelo sistema de forma a criar estatísticas úteis. Vão ser utilizadas as velocidades para saber o sentido das viaturas (velocidade negativa e positiva referem-se a sentidos diferentes) e para calcular as velocidades médias de 10 em 10 minutos. É, também, criado um histórico de velocidades e vai ser guardada a velocidade máxima e mínima desde sempre. Vão ser utilizados os IDs dos objetos para fazer a contagem das viaturas de cada um dos sentidos, calculando também a sua diferença, dando a indicação de quantas viaturas estão naquela zona. |

#### Diagrama de Atividades



Figura 3 - Diagrama de Atividades Estatísticas de Tráfego

### [Configuração Sistema #2]

Tabela 3 - Caso de Utilização #2 - Configuração Sistema

| Nome: |  |
| --- | --- |
| Atores: | Administrador(inicia) |
| Prioridade: | Alta |
| Finalidade: | Configuração da aplicação para cada radar, escolhendo que dados vão ser exibidos. |
| Requisitos funcionais: |  |
| Pré-condições: | Possuir aplicação e interfaces criadas, incluindo uma interface de configuração. |
| Sumário: | O administrador configura o sistema para cada radar, podendo usar esta aplicação independentemente de qual o radar e de qual a sua localização. Administrador tem de indicar quantos radares existem e onde estão, quantas vias existem para cada um dos sentidos. Feita a definição da localização do radar, configura quais as estatísticas que vão estar disponíveis para cada um dos tipos de utilizadores da aplicação. |

#### Diagrama de Atividades

Figura 4 - Diagrama de Atividades Configuração Sistema



### [Consultar informações do tráfego #3]

Tabela 4 - Caso de Utilização #3 - Consultar Informações do Tráfego

| Nome: |  |
| --- | --- |
| Atores: | Condutores e Entidades registadas (inicia) |
| Prioridade: | Alta |
| Finalidade: | Consultar a interface principal da aplicação. |
| Requisitos funcionais: |  |
| Pré-condições: | Interfaces criadas, sistema a gerar estatísticas e configuração do sistema feita. |
| Sumário: | Utilizador abre a aplicação. Sistema mostra a interface principal, que corresponde à interface com as informações para os condutores, utilizadores sem registo, ou seja são as estatísticas menos detalhadas. No caso dos condutores não farão mais nada, no caso das entidades registadas podem prosseguir à autenticação. |

### [Autenticação #4]

Tabela 5 - Caso de Utilização #4 - Autenticação

| Nome: |  |
| --- | --- |
| Atores: | Entidades Registadas (inicia) |
| Prioridade: | Alta |
| Finalidade: | Permitir o acesso a outras informações que utilizadores sem login não podem ver. |
| Requisitos funcionais: |  |
| Pré-condições: | Entidades têm de fazer um registo prévio contactando o administrador diretamente. |
| Sumário: | Utilizador abre a aplicação. Sistema mostra a interface principal. Utilizador escolhe a opção de fazer login. Sistema mostra a interface de autenticação. Utilizador insere as credenciais. No caso do administrador, ele é redirecionado para a sua interface própria de configuração, se for uma entidade registada, mostra a interface destes utilizadores, que possui mais informações estatísticas que a do condutor. |

## Cobertura de requisitos

[Casos de utilização x Requisitos funcionais: para cada C.U., quais os requisitos que ele implementa?]

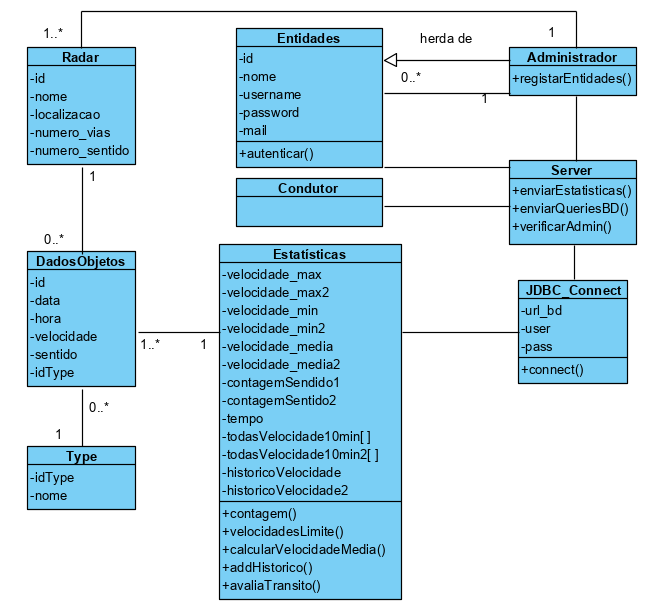
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | RF1 | RF2 | RF3 | RF4 | RF5 | RF6 | RF7 |
| Estatísticas de Tráfego #1 | X | X | X | X |  |  |  |
| Configuração Sistema #2 |  |  |  |  | X | X | X |
| Consultar informações do tráfego #3 | X | X | X | X | X |  |  |
| Autenticação #4 |  |  |  |  |  | X | X |

# Diagrama de Classes

A partir do diagrama de casos de utilização, definiram-se os principais conceitos deste sistema e elaborou-se o diagrama de classes (figura 5). Para cada classe definiram-se os atributos e métodos que fossem necessários a implementar.

Algumas destas classes seriam apenas classes de dados, ou seja, seriam implementadas na base de dados, mas não tinham necessariamente de ser criadas como classes netbeans, serão apenas para armazenar dados. Além disso, também existem alguns métodos que só serviram para a implementação na base de dados.

Figura 5 - Diagrama de Classes



# Diagrama de Entidades

A partir do diagrama de classes foi criado o diagrama de entidades, para a implementação da base de dados da aplicação (figura 6).



Figura 6 - Diagrama de entidades

Para a criação deste diagrama teve-se em conta quais as classes em que

# Implementação da Base de Dados

kjskdjsk

## Criação das tabelas

## Criação das views

## Filtração e inserção dos dados do radar na base de dados

# Principais Métodos a Implementar

* sendEstatisticas()

Este método vai mostrar as estatísticas aos clientes.

Tem de mostrar os dados do radar, mostrar mostrar o histórico, mostrar as estatísticas de cada um dos sentidos.

Vai à base de dados e retira de lá os dados.

Recebe a conexão com o cliente e retorna os dados.

Teste unitário: Em vez de ir verificar às tabelas da BD verifica em tabelas dummy e não recebe uma conexão socket do cliente.

* autenticarUser(String user, String pass)

Este método recebe o user e a pass, vai verificar se existe na bd.

Se existir e coincidir retorna true, se não retorna false.

Teste unitário: Em vez de ir verificar às tabelas da BD verifica em tabelas dummy.

* checkAdmin(String user, String tipo)

Este método recebe o user e o tipo de entidade, e vai verificar se é um administrador ou não.

Retorna true se for administrador e false se não for.

Teste unitário: Em vez de ir verificar às tabelas da BD verifica em tabelas dummy.

* registarEntidades(String nome, String user, String mail, String type)

Este método fará o registo de novas entidades, inserindo-as na Base de Dados.

Recebe os dados da entidade a registar.

Cria uma pass gerada automaticamente.

Coloca o user e a pass na tabela login da Base de Dados e coloca o nome, user, mail e tipo na tabela da entidades da BD.

retorna true se a entidade for inserida na BD e false caso não seja.

Teste unitário: Em vez de inserir em tabelas na BD insere em tabelas dummy.

* exQuery(String query)

Este método recebe uma string que será a query, envia-a à base de dados e não irá retornar nada

Não se fazem testes unitários

* connect()

Faz a conexão com a BD

Não tem parâmetros, mas vai retornar a ligação com a base de dados

Não se fazem testes unitários

# Testes Unitários

Testes de Unidade ou teste unitário é a fase de testes onde cada unidade do sistema é testada individualmente. Unit, é o nome genérico para qualquer estrutura de testes automáticos unitários. Consiste na verificação da menor unidade do projeto de software.

O objetivo é isolar cada parte do sistema para garantir que elas estejam funcionando conforme especificado. Através da utilização deste tipo de teste ao longo da implementação é possível reduzir a quantidade de bugs na aplicação final. Os testes do Unit funcionam através de comparação de resultados das funções a serem testadas com valores esperados.

Neste caso não seria apenas necessário criar testes só com o JUnit pois após algumas pesquisas na internet foi descoberto que como para a classe “JDBCConnect” estávamos a fazer uma conexão à base de dados, na altura de fazer os testes dava um erro, pois como a base de dados está sempre em atualização não é possível fazer os testes porque derá sempre erro, foi encontrado um solução “JMock”.

JMock é uma biblioteca que auxilia o Test Driven Development (TDD) através dos mock objects. É uma biblioteca que vai criar implementações falsas específicas para os testes que queremos testar. Mas para a utilização do JMock é necessário criar uma pasta Maven para criar a biblioteca necessária para ter as funções necessárias todas. Após muitos testes o grupo não conseguiu usar o JMock devido ao projeto Maven, decidimos procurar outra forma tais como “HSQLDB”, “DBUnit”, “NBUnit” mas todos sem sucesso.

# Implementação da aplicação no netbeans

## Aplicação Cliente – Servidor

## Conexão à base de dados – JDBC

Java Database Connectivity ou JDBC é uma interface de programação de aplicativos (API) que reúne conjuntos de classes e interfaces escritas na linguagem Java. Nesta linguagem, o JDBC possibilita a conexão, através de um driver específico, ao banco de dados desejado.

Neste driver pode-se executar instruções SQL de qualquer tipo de banco de dados relacional, com estas instruções é possível manipular instruções do tipo “Select”, “Insert”, “Update”, ”Delete”, “Create” e “Drop” sendo possível invocar “Store Procedures”, que são funções ou procedimentos que ficam armazenados no SGBD escritos numa linguagem própria, através do JDBC. Além deste driver atuar como uma interface entre os Sistema de Gerenciamento de Banco de Dados (SGBD) e as aplicações, também pode ser considerado como um tradutor que ajuda na definição das mensagens binárias trocadas com um SGBD.

Para desenvolver uma aplicação baseada em uma especificação JDBC é preciso entender algumas das classes e interfaces relacionadas ao pacote a ser usado na altura do desenvolvimento, neste caso temos um pacote designado “java.sql” que é utilizado no driver JDBC do banco de dados em MYSQL, SQL Server, DB2 PostgresSQL, entre outros. Este pacote “java.sql” oferece à biblioteca Java o acesso e procedimento de dados tais como “Aplicações Java”, “Applets Java”, “Servlets Java”, “Java ServerPages(JSPs)” e “Enterprise JavaBeans(EJBs)”, os recursos fornecidos por estas bibliotecas são:

* Carregamento automático de driver do banco de dados;
* Execução ao lidar com melhorias;
* Funcionalidade aprimorada de BLOB/CLOB;
* Aprimoramentos na interface de conexão e instrução;
* Suporte ao conjunto de caracteres nacionais;
* Acesso ao SQL;
* Suporte ao tipo de dados em xml.

Um conjunto de bibliotecas básicas da plataforma Java contém classes como “Object”, “String” e “Date”, coleções como “List” e “Set”, além das classes para a manipulação de arquivos, redes e interfaces gráficas, entre outras funcionalidades básicas da plataforma de desenvolvimento. Para a comunicação com a base de dados é necessário obter arquivos com a extensão “.jar” de modo a ajudar no carregamento do driver do JDBC.

Posto isto e com a ajuda do JDBC, vai-se definir, como um cliente pode ter acesso a esse mesmo banco de dados. É uma tecnologia de acesso usada para conectividade de banco de dados em Java.

## Criação interfaces em JFrame

## Implementação classes e métodos

### Mostrar estatísticas simples

### Mostrar histórico

### Autenticação

### Registo de Entidades

# Análise dos Resultados

REQUISITOS CUMPRIDOS E NÃO CUMPRIDOS

OBJETIVOS CUMPRIDOS E NÃO CUMPRIDOS

# Reflexão Crítica e Conclusão

No geral, foram cumpridos a maioria dos objetivos e dos requisitos definidos inicialmente, apesar de certas funcionalidades que tinham sido inicialmente planeadas terem sido descartadas, com o intuito do grupo se focar no que realmente era importante. Definiu-se o que tinha a maior prioridade e foi isso que foi implementado.

Em relação ao planeamento do projeto, como se optou por uma abordagem ágil, o desenvolvimento do projeto deveria ser dirigido por testes. Contudo, durante as fases iniciais, o grupo teve algumas dificuldades no planeamento, o que gerou alguma confusão para os membros do grupo. Assim, como o grupo demorou um pouco mais do que devia a perceber o que tinha de ser implementado e como, acabou também por sofrer um pequeno atraso no desenvolvimento do projeto. Mesmo assim, o grupo conseguiu ultrapassar essas dificuldades, definir prioridades, refinar os diagramas e seguir em frente.

Quando se chegou à fase da criação dos casos de testes, estes não foram bem definidos, não sendo logo feitos no JUnit, isso só foi feito posteriormente. Definiram-se quais os principais métodos que tinham de ser implementados no NetBeans, quais os seus parâmetros e o que tinham de retornar. Essa definição foi o suficiente para dar ao grupo uma noção do que tinha de ser implementado, passando então para a parte da programação.

Apesar da criação da Base de Dados não ter gerado muitos problemas, o código foi na mesma refactorizado várias vezes, pois iam sendo encontrados alguns problemas na criação das tabelas e das views.

A implementação da aplicação no NetBeans, foi definitivamente das partes mais complicadas do projeto. A implementação das várias classes foi sendo feita por partes e demorou algum tempo até tudo estar em ordem.

No que toca aos objetivos deste projeto, planeava-se conceber uma aplicação que:

1. permitisse a gestão do tráfego para qualquer zona à escolha do administrador;
2. permitisse a qualquer condutor obter estatísticas do tráfego;
3. permitisse a certas entidades o acesso a informações mais detalhadas do trânsito.

Os objetivos 2 e 3 eram os que tinha prioridade e foram cumpridos. Relativamente ao objetivo 1, este não foi cumprido, uma vez que esta aplicação só permite a gestão de apenas um radar e o administrador não consegue configurar o sistema adicionando novos radares.

Além disso, simplificaram-se as interfaces do utilizador, inicialmente pensou-se em ter um mapa que seria manipulável, mas optou-se pela criação de interfaces mais simples apenas mostrando as estatísticas.

dkflklaflkald

Em conclusão, slakslskla

# Anexos

[se necessário documentação deve ser anexada e convenientemente indexada de modo a ser referenciada no documento]

[listar ficheiros externos, nomeadamente os ficheiros.mdl a consultar]