

Sistema Brooklyn 99

Sistema de Registro e Direcionamento de Viaturas Policiais

Lucas Borges
Mírian Francielle
Tatiana Camelo

1. Introdução

O departamento de polícia do Brooklyn, precisa de um sistema que permita que incidentes sejam registrados e direcionados ao veículo policial mais apropriado. Os fatores que devem ser levados em conta na hora de decidir para qual veículo enviar cada incidente são:

(i) o tipo de incidente, pois alguns incidentes são mais sérios que outros e necessitam de uma resposta mais urgente;

(ii) a posição de veículos disponíveis, pois, em geral, a melhor estratégia é solicitar que o veículo mais próximo atenda ao incidente. Entretanto, deve ser considerado que a posição dos veículos pode não ser conhecida exatamente, sendo necessário solicitar ou estimar a posição dos veículos.

Além disso, deve-se considerar:

(iii) os tipos de veículos disponíveis uma vez que alguns incidentes requerem uma série de veículos, enquanto outros podem requerer veículos especializados;

(iv) a localização do incidente, pois em algumas áreas pode ser imprudente enviar um único veículo. Finalmente, o sistema deve também levar em conta

(v) a necessidade de alertar outros serviços de emergência, tais como serviços de bombeiro e ambulância. Se necessário, o sistema deve alertar automaticamente esses serviços. Todos os dados de incidentes devem ser relatados por um relator que pode ser o policial que atendeu ao chamado.

2. Planejamento Pré Definido

- ❑ Modelo a ser utilizado: Espiral com Prototipação;
- ❑ Arquitetura em camadas;
- ❑ Implementação de funcionalidades e Integração com API;
- ❑ Testes para Liberação de Protótipo;
- ❑ Implementação de sugestões e modificações necessárias;
- ❑ Entrega final do Software com todo o time.

3. Cronograma de Atividades do time



Project Schedule

SISTEMA BROOKLYN 99

Gabriel Fanelli
Lucas Borges
Mirian Francielle
Tatiana Camelo

WEEK 1

Mirian (01/10 - 03/09)

- Análise e Definição de Requisitos.
- Especificação de Necessidades.

Tatiana (04/10 - 06/10)

- Estudo de Viabilidade.
- Documentação e Validação de Requisitos.

WEEK 2

Mirian (08/10 - 12/10)

- Projetar Arquitetura
- Implementação das funcionalidades

Tatiana (13/10)

- Validação das Funcionalidades

WEEK 3

Lucas (15/10 - 17/10)

- Integração com API do sistema de Delegacias

Mirian, Lucas (18/10 - 20/10)

- Teste Unitário para assegurar requisitos.

WEEK 4

Lucas (22/10 - 24/10)

- Teste de Integração

Mirian, Lucas (25/10 - 27/10)

- Liberação de Protótipo

WEEK 5

Time (29/10 - 31/10)

- Implementação de Sugestões e Modificações
- Implantação

Time (09/11)

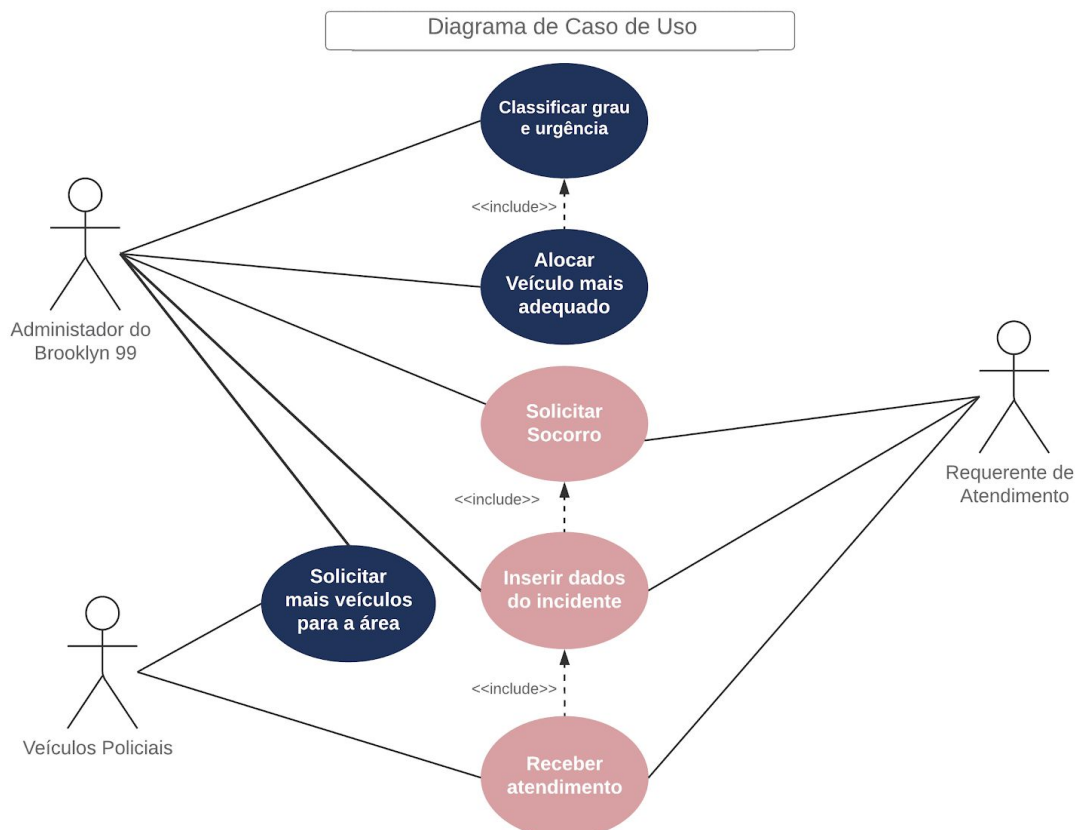
- Entrega Final do Sistema

4. Requisitos do Sistema

De acordo com as especificações do departamento de polícia, suas respectivas necessidades e estudos de viabilidade. Segue-se os requisitos documentados e validados com o time:

- ❑ O sistema deve ser capaz de armazenar uma lista de incidentes à medida que os mesmos ocorrem;
- ❑ O sistema deve manter uma lista de veículos policiais associados ao mesmo;
- ❑ O sistema deve ser capaz de informar ou estimar a posição atual de um veículo policial;
- ❑ O sistema deve coletar dados do incidente diretamente da chamada realizada pelo solicitante;
- ❑ Os incidentes devem ser classificados em grau de intensidade e urgência;
- ❑ Os veículos devem conter tipos e especializações;
- ❑ O sistema deve ser capaz de alocar o veículo mais adequado (distância e especialização) para um dado incidente;
- ❑ O incidente deve ser avaliado pela área de ocorrência, em que áreas de risco recebem mais veículos;
- ❑ O veículo selecionado para o atendimento, deve ser capaz de solicitar mais veículos e contactar outros sistemas de emergência;
- ❑ O requerente do serviço, deve ser capaz de inserir as informações correspondentes ao incidente no sistema.

O diagrama de casos de uso abaixo, apresenta o comportamento do sistema através de perspectivas do usuário, como uma modelagem das funcionalidades principais dos requisitos do sistema.



Cenário do Caso de Uso

Nome do cenário: Envio de viaturas para acidente

Ator: Administrador, policiais, requerentes de atendimento

Pré-condição: Acidente Ocorrido

Fluxo normal:

1. Ocorrência de acidente
2. Solicitação de atendimento
3. Inserção de dados do acidente
4. Policial administrador do sistema faz classificação de grau e urgência
5. Alocação de veículo mais adequado
6. Veículo policial é enviado ao local
7. Atendimento é realizado.

Fluxos alternativos:

6.1. Não há policiais e viaturas que atendem os pré-requisitos

6.2. São solicitados novos veículos que se adequem a necessidade do acidente

Pós-condição: Viatura condizente é enviada para atendimento de socorro e solicita outro sistema de emergência.

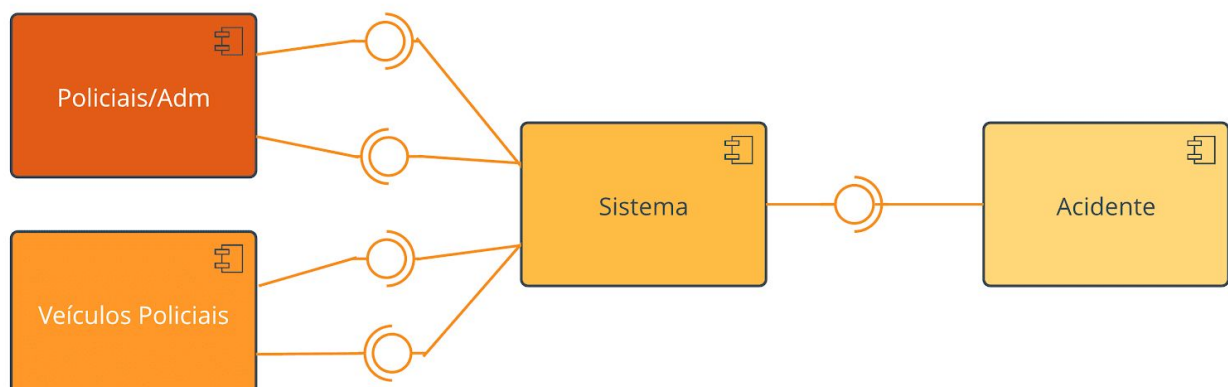
5. Projeto Arquitetural

A arquitetura deve considerar requisitos não funcionais:

- ❑ Desempenho: evitar comunicação excessiva entre componentes distribuídos.
- ❑ Segurança: ocultar características críticas de segurança em camadas mais internas.
- ❑ Disponibilidade: incluir componentes redundantes e tolerância à falhas.

O diagrama de componentes abaixo, identifica os principais subsistemas que serão desenvolvidos, e como eles se comunicam. Os administradores do Sistema podem se comunicar diretamente com o sistema, do mesmo modo que o sistema pode fornecer dados aos policiais administradores. Os veículos policiais também se comunicam com o sistema diretamente, dada as disponibilidades de carros e tipos, assim como o sistema se comunica com os carros. E o acidente, apenas fornece dados para o sistema, que repassa aos veículos e aos administradores.

Diagrama de Componentes



5.1 Modelo de Projeto

O projeto utilizado combina elementos dos modelos Incrementais e de Prototipagem, e sequência adotada do Modelo Cascata, de tal modo que é desenvolvido em versões: Prototipagem nas primeiras versões e Incremental nas últimas versões. A escolha por esse modelo, é justificada por ele ser do tipo que lidam com mudanças, também conhecidos como modelos evolucionários, por serem iterativos e possibilitarem o desenvolvimento de versões cada vez mais completas.

O presente software documentado, utilizará os dois modelos mais comuns desta categoria: Prototipação e Espiral. Como o software evolui com o tempo e de acordo com as necessidades do departamento, essa escolha se torna mais adequada para o projeto de desenvolvimento do Sistema de Controle de Veículos Policiais.

5.2 Arquitetura em camadas

A arquitetura do Sistema, é definida em camadas, para melhor distribuir os serviços e utilidades do sistema, como foco na qualidade e eficiência. Todo o sistema de informações, solicitações, gerenciamento e alocação, é executado em um único computador, as camadas intermediárias estão implementadas como um único programa que se comunica com o banco de dados por meio de sua API.



User Interface

Resource management

Application Functionality/ System utilities

System Support

❑ *System Support:* A camada mais baixa da arquitetura, inclui o banco de dados com a distribuição de localização de todos os veículos policiais e o gerenciamento da transação. Que inclui recursos que bloqueiam a duplicação de aplicação de alocação de veículos para usuários que fazem solicitações simultâneas.

❑ *Application Functionality/System utilities:* Camada de aplicação que inclui os componentes relacionados com as funcionalidade da aplicação/controle do departamento. Alocação de recursos que atualiza o banco de dados de solicitações, quando os recursos são atribuídos, associa-se aos dados do solicitante.

❑ *Resource management:* A camada seguinte, inclui a classificação do incidente de acordo com os dados do usuário que solicitou, para a realização do atendimento eficiente.

❑ *User Interface:* Componente de interface do usuário, onde é feito a solicitação de socorro/atendimento. Nesse sistema, a solicitação pode ser feita por uma chamada.

5.3 Diagramas Comportamentais e Estruturais

A administração do sistema B99, é feita por policiais autorizados do departamento. Ao receberem os pedidos de atendimento, realizam a busca da localização exata do acidente e dos veículos mais próximos. De acordo com os dados fornecidos pelo requerente, a administração classifica o grau de prioridade e avalia o risco do envolvido, com essas informações estabelecidas, o sistema pode alocar o veículo mais próximo que é adequado para o atendimento.

Ao chegar, o veículo realiza mais uma avaliação, podendo ou não solicitar ao administrador reforços e/ou contactar outros serviços de emergência. Em seguida, é feito o atendimento. Dado esse modelo arquitetural, segue-se os **diagramas comportamentais** do software.

Diagrama de Atividades

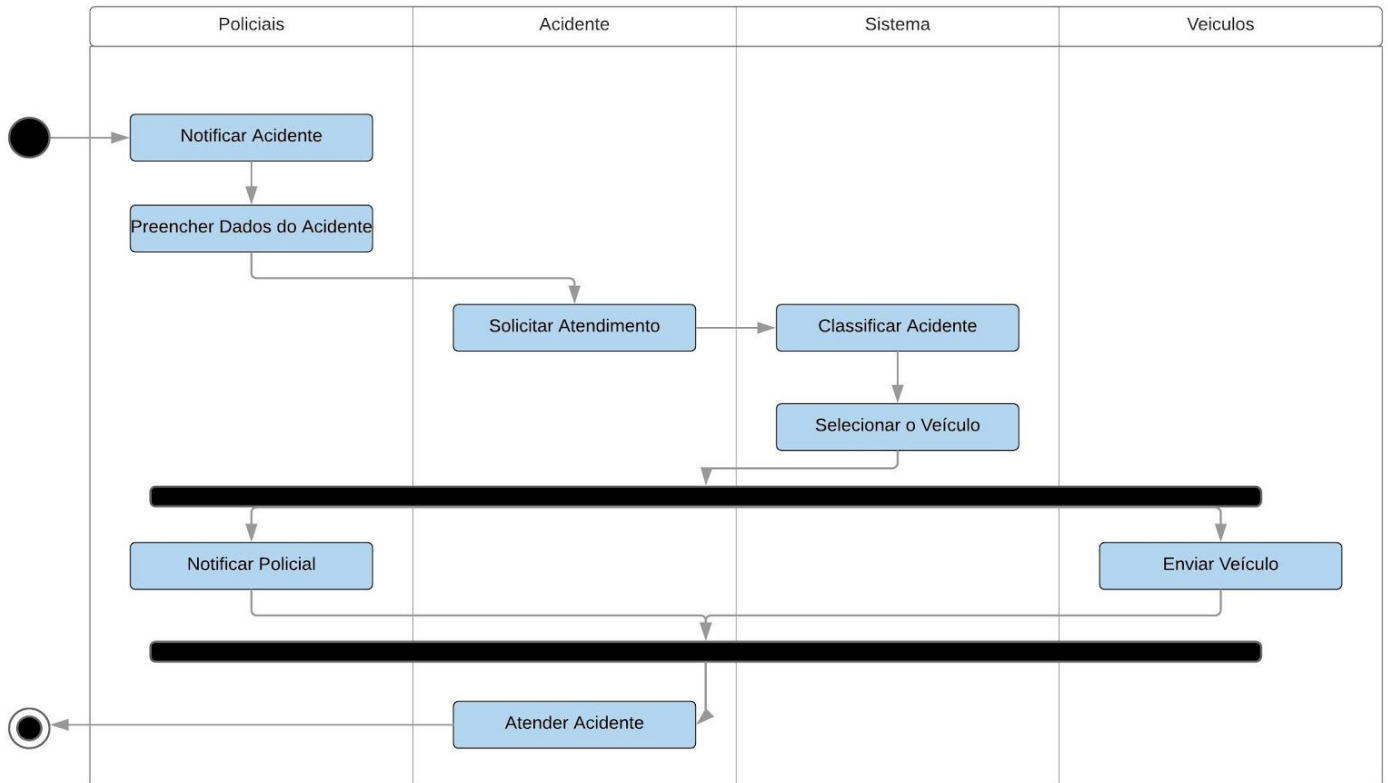
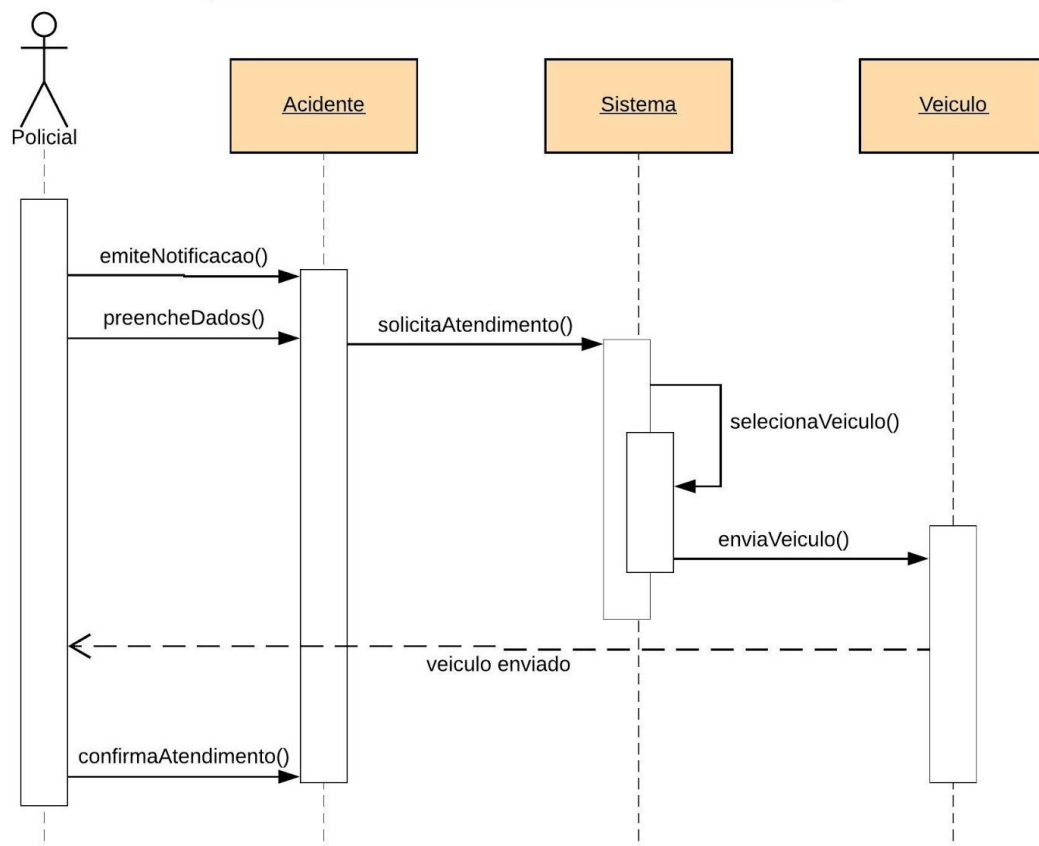
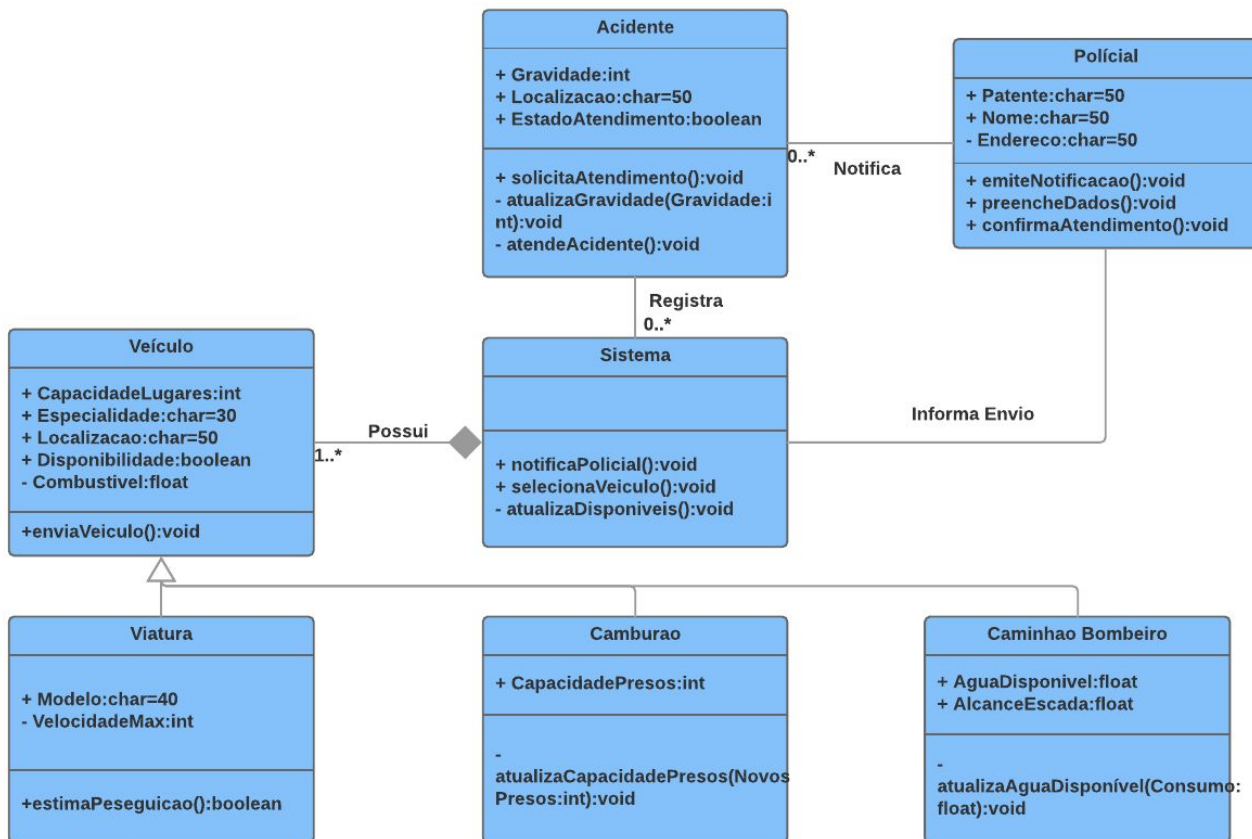


Diagrama de Sequência



Dados os diagramas comportamentais do sistema, segue-se o **diagrama estrutural** das classes do software. Cada classe com seus atributos e visibilidades, seguidos dos respectivos métodos e visibilidades.

Diagrama de Classes



Ferramentas utilizadas:

- GitHub (Repositório): <https://github.com/mirianfsilva/brooklyn99/>
- IntelliJ IDEA 2018.2.4
- Lucidchart (Diagramas e modelos UML): <https://www.lucidchart.com/>