



Sistema Integrado de Controle de Veículos

SICV

1ª Entrega Projeto Engenharia de Software

1- Membros da Equipe:

- a. Carlos Henrique Maciel Sobral Timoteo
- b. Diego Albuquerque de Araújo
- c. Eliackin Messias do Nascimento Figueiredo
- d. Rafael Praxedes Gomes

2- Descrição do Sistema:

Sistema Integrado de Controle de Viaturas - SICV

O Sistema é baseado na aplicação da Tecnologia da Informação na área de Segurança Pública e Infra-estrutura de Serviços. O CIODS (Centro Integrado de Operações de Defesa Social) tem um sistema que centraliza as ocorrências e as repassa para as subunidades interligadas a ele (PM, PC, CBC, PF...). Ele funciona da seguinte maneira: o solicitante liga para Central do CIODS, um atendente recebe a solicitação e repassa para um despachante de uma subunidade e este delega os atuadores para prestar os serviços. O SICV propõe automatizar as atividades de selecionar a viatura mais apropriada para a ocorrência, sugerir a melhor rota, estabelecer prioridades das ocorrências e monitorar o atendimento, ou seja, auxiliando o trabalho do despachante.

O SICV é uma aplicação web composta por três módulos, um acessado pela viatura, outro pelo despachante e o último pelo atendente.

Os atuadores/viaturas se encarregarão de informar a sua posição mediante o acontecimento de um evento disparado pela Central. Cada atuador terá um Sistema de Posicionamento e terá acesso ao Mapa da Região onde ele se encontra. Além disso, ele se comunicará com a Central via Internet para poder requisitar serviços como: gerar relatório policial, enviar mensagem para a central, chamar mais viaturas para a ocorrência, receber informações sobre a ocorrência da central e outros.

O despachante recebe a ocorrência de um atendente, o SICV localiza a melhor viatura para a ocorrência e envia sugestão da melhor rota, para só assim estabelecer um vínculo **de monitoramento** entre o despachante e a viatura.

O atendente preencherá o formulário da ocorrência com os dados obtidos do solicitante e enviará a ação diretamente para o SICV(Módulo Despachante) da Central.

O SICV será: uma aplicação distribuída para atender uma grande demanda de solicitações garantindo assim a escalabilidade, desenvolvida na plataforma Java por ser portátil e ser uma tecnologia conhecida pelos desenvolvedores do sistema e desenvolvido em "Camadas" para garantir maior modularidade. O sistema **também** terá um sistema de log e controle de concorrência.

3- Histórias de Usuário:

1ª História: Representar o mapa de uma região no sistema.

Prioridade: Alta

2ª História: Localizar uma viatura no mapa.

Prioridade: Alta

3ª História: Determinar a posição de uma ocorrência no mapa.

Prioridade: Alta

4ª História: O sistema irá sugerir a melhor rota para uma viatura que foi acionada.

Prioridade: Média

5ª História: Gerenciar as ocorrências de acordo com as suas prioridades (gravíssima, grave, média, baixa, baixíssima).

Prioridade: Alta

6ª História: Acionar a viatura mais próxima do local de uma ocorrência.

Prioridade: Alta

7ª História: Permitir a comunicação do policial da viatura com o despachante. O policial passará informações sobre o andamento da ocorrência e receberá um acompanhamento do despachante.

Prioridade: Média

8ª História: Armazenar todas as ações de todos os envolvidos na ocorrência para auxiliar futuras investigações.

Prioridade: Baixa

9ª História: Todas as informações relevantes (ocorrências, pessoas envolvidas, entre outras) devem ser armazenadas em uma base de dados.

Prioridade: Média

10ª História: Mostrar um mapa gráfico na tela do despachante para que ele possa acompanhar a viatura em movimento.

Prioridade: Baixa

4- Riscos do Projeto

Risco: Não aprender a tempo como implementar a concorrência do sistema.

Probabilidade: Baixa

Solução: Dedicar mais horas do dia ao seu aprendizado.

Risco: Usar as ferramentas JUnit, Ant e XPlanner.

Probabilidade: Média

Solução: Aprender a usar o JUnit , o Ant e não usar o XPlanner, uma vez que não é obrigatório o seu uso.

Risco: Não aprender, a tempo, a usar a tecnologia GPS.

Probabilidade: Alta

Solução: Dedicar mais horas do dia ao seu aprendizado.

Risco: Não aprender a tempo como implementar a persistência *do sistema*.

Probabilidade: Baixa

Solução: Dedicar mais horas do dia ao seu aprendizado.

Risco: Falta de tempo para se dedicar ao projeto devido a outras disciplinas e a iniciação científica.

Probabilidade: *Média*

Solução: Dedicar mais horas do dia ou da noite ao seu aprendizado.

Risco: A ferramenta utilizada para realizar algumas tarefas já estabelecidas, não for capaz de ser auxiliar em novas atividades relacionadas com as já relacionadas.

Prioridade: Média

Solução: Encontrar novas ferramentas que atendam essa nova necessidade e se possível uma que integrem todas as outras ou a maior parte das que já foram implementadas.

Risco: O não conhecimento do comportamento e da especificação de partes do sistema e implementá-las de uma maneira inviável.

Probabilidade: Média.

Solução: Estudar e pesquisar em fontes confiáveis para se ter um conhecimento básico.

Risco: Haver conflito de horários para programar em pares.

Probabilidade: Alta

Solução: Sacrificar horários não usuais para programar.

Risco: Problemas com gerenciamento as versões do sistema.

Probabilidade: Média.

Solução: Utilizar o SVN.

Risco: Estabelecimento de Cronograma.

Probabilidade: Alta.

Solução: Diminuir o rigor no seguimento do cronograma ou sempre estar preparado para um cronograma alternativo.

Risco: Atraso no Cronograma.

Probabilidade: Alta.

Solução: Arcar com as consequências e sempre ter em mente que foi feito o melhor possível, para assim nunca abalar a moral da equipe.

5- Implementação das historias de Usuário

A 1ª História foi implementada pois as outras histórias dependiam de sua implementação. Ela foi implementada por Rafael e Carlos Henrique.

A 2ª e 3ª a História foram implementadas por serem partes importantes para a implementação da principal funcionalidade do sistema que e localizar a melhor viatura para atender uma determinada ocorrência. Ela foi implementada por Elliackin e Diego.

6- Lista de Ferramentas:

- Linguagens:
 - 1 – Java 6.0
- Compiladores:
 - 1 – javac (Java Compiler).
- Ambientes de Desenvolvimento:
 - 1 – Eclipse (vs3.1 / vs3.2 / vsEuropa)
 - 2 – NetBeans 6.0
 - 3 – JUDE 3.0.2
- Arcabouços de Testes:
 - 1 – JUnit 4.8 at Netbeans 6.0.
 - 2 – JUnit 3.8.1 at Eclipse.
- Ferramentas para produção de builds:
 - 1 – Ant Builder.
- Ferramentas para o registro de histórias de usuário:
 - 1 – MS Word.
- Ferramentas para dar suporte à escrita de histórias de usuários:
 - 1 – MS Word.
 - 2 – XPlanner.