



Sistema Integrado de Controle de Veículos  
**SICV**

4ª Entrega Projeto Engenharia de Software

Recife, 09 de junho de 2008

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 as 5 melhores viaturas e suas rotas para atender uma ocorrência.

*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 escolhida pelo agente.

*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:

##### *Riscos Imutáveis:*

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

*Probabilidade:* Baixa

*Severidade:* Média

*Solução:* Restringir a escalabilidade do sistema.

*Situação:* Estamos implementando isso nessa iteração.

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

*Probabilidade:* Baixa

*Severidade:* Alta

*Solução:* Utilizar arquivos e memória para armazenar os dados.

*Situação:* Nós estamos implementando a persistência do sistema.

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

*Probabilidade:* Alta

*Severidade:* Alta

*Solução:* Administrar melhor o tempo e fazer horas extras.

*Situação:* Esse risco comprometeu de forma absoluta o desenvolvimento do sistema, já que tivemos uma série de provas, projetos a serem entregues, além da Iniciação Científica.

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

*Prioridade:* Média

*Severidade:* Média

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

*Situação:* Ainda existe, já que por exemplo o Eclipse não dá suporte à verificação de arquivos JavaScripts.

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

*Probabilidade:* Média.

*Severidade:* Alta

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

*Situação:* Estamos continuamente enfrentando esse risco, nessa etapa final.

*Risco:* Haver conflito de horários para programar em pares.

*Probabilidade:* Alta

*Severidade:* Baixa

*Solução:* Sacrificar horários não usuais para programar.

*Situação:* Devido ao fato de todos os elementos do grupo estar envolvidos com outras atividades também muito importantes como Trabalho e Iniciação Científica, houve diversos conflitos de horário.

*Risco:* Problemas com gerenciamento das versões do sistema.

*Probabilidade:* Média.

*Severidade:* Média

*Solução:* Utilizar o SVN.

*Situação:* Ainda existe do mesmo jeito que no início do desenvolvimento, porque existem vários problemas na união das várias versões de código.

*Risco:* Atraso no Cronograma.

*Probabilidade:* Alta.

*Severidade:* Média

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

*Situação:* Houve esse risco, por conta de vários fatores:

- Estudar para Provas
- Fazer Trabalhos de Disciplinas
- Um integrante do grupo trabalha e Três fazem Iniciação Científica
- Aprender JavaScript, CSS, HTML, Ajax e XML e suas interações.
- Aprender a usar a API do Google.
- Problemas com o uso de Ajax.

Esses fatores podem ocasionar a não entrega de todas as histórias de usuário.

*Risco:* Implementar páginas HTML que usem tecnologia CSS, JavaScript, Ajax e XML.

*Probabilidade:* Alta

*Severidade:* Alta

*Solução:* Recorrer a tutorias, livros e sites na web para aprender as tecnologias e ajudar no desenvolvimento.

*Situação:* O nosso grande desafio é fazer a ligação entre as páginas HTML e o controle e processamento de dados local via Servlets, Ajax e leitura de arquivos XML. Não encontramos um navegador que oferecesse suporte para o uso de Ajax.

*Risco:* Não conseguir usar a API do GoogleMaps como ferramenta acessória do sistema.

*Probabilidade:* Alta

*Severidade:* Alta

*Solução:* Olhar o Tutorial do GoogleMaps API, conceitos básicos e principais da API e Google Maps API Reference para identificar e compreender os métodos e atributos usados pelo código.

*Situação:* Não conseguimos utilizar a biblioteca para lidar com a tecnologia Ajax, assim tivemos que utilizar um código completo de tratamento Ajax, no entanto, os navegadores não responderam positivamente em nenhum dos dois casos.

### *Riscos Mitigados:*

*Risco:* Usar as ferramentas JUnit, Ant e XPlanner.

*Porque:* Foi mitigado porque nós aprendemos a usar essas ferramentas.

*Como:* Tivemos de dedicar algumas horas a manuseá-las, assim desenvolvemos a habilidade de usá-las, de forma intuitiva. No caso do JUnit e Ant, recorremos a tutoriais e sites de ajuda para guiar no aprendizado. Já com o XPlanner, olhando o documentação foi indicado o manuseio direto da ferramenta, já que as suas funcionalidades e características são intuitivas para pessoas que têm certo conhecimento em métodos ágeis.

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

*Porque:* Seria dispendioso trabalhar com a tecnologia GPS, então muitos recursos seriam absorvidos para pouco retorno. Portanto, nós vamos desenvolver uma classe GPSSimulator.

*Como:* Pesquisando em sites, "blogs", tutorias e olhando sites de empresas fabricantes, foi identificado que cada GPS tem a sua API Empresarial de uso, que usa linguagens específicas. Além do mais, nós não tínhamos recursos suficientes para a compra do equipamento. Por isso, decidimos criar um "simulador de GPS".

*Risco:* Mudança nas Escolhas de Projeto e na Arquitetura do sistema.

*Porque:* Definimos e encontramos a melhor solução para o nosso sistema.

*Como:* Estudando pelos livros e realizando pesquisas na internet. Dessa maneira, conseguimos a melhor visão do sistema.

*Risco:* O Servidor do GoogleMaps falhar na construção de partes do sistema.

*Porque:* A probabilidade do servidor do Google cair é praticamente nula.

*Como:* Em todas as nossas atividades não tivemos problema com a "liveness" do servidor.

*Risco:* O Servidor Local não funcionar adequadamente.

*Porque:* Estamos utilizando um servidor que é bem aderido ao Eclipse.

*Como:* Procuramos e nos informamos que o Apache Tomcat 6.0 é uma ótima escolha para quem precisa de um servidor web local.

*Risco:* Não aprender a usar os diagramas UML

*Porque:* Conseguimos desenvolver uma representação satisfatória da descrição dos diagramas.

*Como:* Utilizando o programa gerador de diagramas(JUDE) e pesquisando na internet algumas soluções desenvolvidas.

*Risco:* Não conseguir montar a Arquitetura e as Visões eficientemente.

*Porque:* Desenvolvemos um modelo eficiente do sistema.

*Como:* Estudando nos livros *Software Architecture in Practice*, *Documenting Architecture e Design Patterns*.

*Risco:* Não conseguir documentar a arquitetura do sistema eficientemente.

*Porque:* Lemos o livro Documenting Architecture.

*Como:* Adaptamos alguns exemplos de documentação para a nossa arquitetura.

*Risco:* Não conseguir implementar e usar o banco de dados corretamente.

*Porque:* Estamos realizando a persistência do sistema através de JPA, assim, foi excluído o uso do banco.

*Como:* Pesquisando na internet outras soluções e tendo como fonte inspiradora a solução desenvolvida no *Health Watcher*.

## 5- Implementação e Descrição das Histórias de Usuário

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

*Descrição:* Como essa história é uma das mais básicas e principais ela foi implementada primeiro. Após o uso da Google Maps API essa história se restringiu a inicializar o mapa do Google Maps e configurar as características iniciais de qualquer mapa, como por exemplo setar o centro do mapa. Os riscos envolvidos nessa história são os relativos ao servidor do GoogleMaps.

*Duplas Envolvidas:* Rafael e Diego.

-Tarefa 1: Carlos, Rafael e Diego - Finalização da parte lógica(funcionalidades) das páginas HTML.

2ª História: Localizar uma viatura no mapa.

*Descrição:* A localização da viatura do mapa será feita através de "GMarkers" com o ícone representando a viatura. Isso inclui o fato de que, como as viaturas se movimentarão, os marcadores irão se movimentar no mapa e representarão as posições ocupadas pelas viaturas num determinado momento. Foi utilizado a comunicação de código Java com JavaScript através de documentos XML e requisições Ajax para essa atividade. Os riscos envolvidos são relativos ao servidor do Google e ao servidor local que armazenará o arquivo "xml".

*Duplas Envolvidas:*

- Tarefa 1: Rafael e Carlos – Gerar código de Escrita e Leitura de arquivos XML em Java.

- Tarefa 2: Carlos e Eliackin – Implementar os marcadores com os ícones das viaturas.

- Tarefa 3: Eliackin e Diego – Gerar a classe *util.js* em JavaScript e métodos de uso para poder pegar as posições das ocorrências e realizar todas as ações com as posições como a seleção das melhores viaturas para atender uma ocorrência.



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

*Descrição:* A atendente vai clicar no mapa o local onde ocorreu a ocorrência. Após isso a posição selecionada vai ser passada para o atendente.

*Duplas Envolvidas:*

- Tarefa 1: Carlos e Diego - Representar a localização da ocorrência no mapa.
- Tarefa 2 : Diego e Elliackin - Leitura da posição e envio de dados para o servidor através do *action* para um Servlet.
- Tarefa 3: Carlos e Rafael - Uso de Sevlets para a comunicação das páginas HTML com a Fachada.
- Tarefa 4: Carlos e Diego - Uso do padrão Commands para executar as ações das páginas HTML.

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

*Descrição:* Após o atendente receber a posição de uma ocorrência, o sistema vai sugerir as melhores viaturas e ele vai escolher e encaminhá-la.

*Duplas Envolvidas:*

- *Tarefa 1:* Elliackin e Diego - Sugerir as 5 viaturas, gerar suas rotas e ordená-las.
- Tarefa 2: Rafael e Elliackin - Mostrar as viaturas sugeridas para o atendente escolher.
- Tarefa 3: O atendente irá escolher uma viatura e enviará os dados.

*Situação:* Não implementada.

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

*Descrição:* Criar uma classe que implemente essa característica das ocorrências.

*Duplas Envolvidas:* Carlos e Elliackin.

6ª História: Acionar a viatura escolhida pelo agente.

*Descrição:* Ao clicar numa viatura e fazer uma chamada, a viatura vai receber do agente os dados da ocorrência e vai receber a rota.

*Duplas Envolvidas:* Diego e Elliackin.

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.

*Descrição:* Fazer a comunicação com a viatura escolhida. Foi usado um Gadget do G!Talk para fazer isso.

*Duplas Envolvidas:* Carlos e Rafael.

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

*Descrição:* A cada comunicação vai ser armazenado num arquivo ou banco de dados as ações dos envolvidos como uma cópia dos métodos executados.

*Duplas Envolvidas:* Diego e Elliackin.

*Tarefa 1:* Uso de JPA para implementar a persistência.

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

*Descrição:* Os objetos que representam entidades reais do sistema estarão armazenados um banco de dados, com as seguintes tabelas: Ocorrência e *Employee*(*Attendant*, *ForwardinAgent* e do Policial da Viatura ).

*Duplas Envolvidas:* Diego e Elliackin

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

*Descrição:* O despachante terá na sua página "html" um GoogleMap em que ele acompanhará e visualizará as viaturas no mapa.

*Duplas Envolvidas:* Carlos e Rafael – Rafael e Diego

Tarefas Extras já realizadas:

1. Uso do Ant para a compilação do Sistema.
  - a. Diego

*OBS.: A Arquitetura do Sistema se encontra no documento "Projeto da Arquitetura.pdf".*