

---

# DESENVOLVIMENTO DE APLICAÇÃO PARA CONTROLE DE ATENDIMENTOS DE EMERGÊNCIA UTILIZANDO PROGRESSIVE WEB APP (PWA)

Caetano F. B. da Silva<sup>1</sup>

Fredson Costa<sup>2</sup>

## RESUMO

O Corpo de Bombeiros Militar do Tocantins trabalha diuturnamente no atendimento de emergências fora do ambiente hospitalar, o que compreende acidentes de trânsito, emergências clínicas, incêndios, buscas e salvamentos terrestres ou aquáticos dentre muitas outras. Este atendimento precisa ser realizado no menor tempo e com a maior qualidade possível tendo em vista sua natureza. Por essa característica é imprescindível que o militar foque seus esforços no atendimento propriamente, em detrimento da necessidade de colher todas as informações relacionadas ao atendimento. Como solução para esse problema, este trabalho aborda o desenvolvimento de uma aplicação multiplataforma para controle de emergências, capaz de capturar grande parte dos dados de forma autônoma, utilizando para isso tecnologias disponíveis em dispositivos móveis. Nesse sentido, será utilizado a tecnologia conhecida por Progressive Web App (PWA).

**Palavras-chaves:** Atendimento de emergência. Corpo de bombeiros. Multiplataforma. Aplicação. PWA.

## ABSTRACT

The Tocantins Military Fire Department works day and night in emergency care outside the hospital environment, which includes traffic accidents, clinical emergencies, fires, land and water searches and rescue, among many others. This service needs to be carried out in the shortest time and with the highest quality possible given its nature. Because of this characteristic, it is imperative that the military focuses its efforts on attendant properly, to the detriment of the need to collect all information related to care. As a solution to this problem, this work addresses the development of a cross-platform application for emergency control, capable of capturing much of the data in an autonomous way, using technologies available on mobile devices. In this sense, the technology known as Progressive Web App (PWA) will be used.

**Keywords:** Emergency care. Fire department. Cross-platform. Application. PWA.

## 1 INTRODUÇÃO


É notório o quão rápido a tecnologia da informação vem se desenvolvendo, afetando diretamente o comportamento das pessoas. Do ponto de vista das instituições, muitas vezes, esse desenvolvimento reflete mudanças ou até mesmo a substituição completa de suas soluções.

Com relação aos dispositivos utilizados, estes estão cada vez mais portáteis, móveis e inteligentes em detrimento do que eram há alguns anos atrás.

---

<sup>1</sup>Aluno do curso de Pós-Graduação em Desenvolvimento de Softwares para Dispositivos Móveis da Faculdade Católica do Tocantins – E-mail: caetano.burjack@gmail.com

<sup>2</sup>Professor do curso de Pós-Graduação em Desenvolvimento de Softwares para Dispositivos Móveis da Faculdade Católica do Tocantins – E-mail: fredson@catolica-to.edu.br



---

Segundo o Sistema Interno de Estatísticas do Corpo de Bombeiros Militar do Tocantins (CBMTO), a instituição presta em média, 11 mil atendimentos por ano e todos esses são registrados, para ao final, geram estatísticas, extratos para os usuários e dados para gestão.

Porém, o sistema e o processo utilizados foram criados num momento em que os dispositivos móveis não eram populares, e o foco era o desenvolvimento de sistemas para internet e uso essencialmente em computadores. Este fato, hoje, abre espaço para uma gama de possibilidades de se aprimorar algumas das soluções existentes no CBMTO e otimizar o uso dos recursos atualmente disponíveis.

Nesse sentido, este trabalho explora a possibilidade de aperfeiçoar um ponto específico do processo de registro do atendimento. Trata-se do acompanhamento da viatura, que se estende do momento em que é acionada, até o seu retorno à unidade.

Atualmente, esse acompanhamento ocorre de forma que a central de emergências, onde se recebem os chamados, se comunica através de rádios transmissores ou telefone com a unidade que deverá prestar o atendimento. Cada informação da viatura é colhida por um dos militares em serviço nesta, e informado à central através dos meios supracitados. Todos os dados, tais como, informação das quilometragens (hodômetro) e horários em cada ponto em que a viatura iniciar e terminar um deslocamento são passados e confirmados. Depois disso, são inseridos, pela central de atendimento, no sistema atualmente em uso.

Os maiores problemas enfrentados nesse processo são: a necessidade de tomar a atenção do motorista da viatura para a passagem do hodômetro; possíveis disparidades em relação ao horário enviado e o registrado no sistema; dificuldades para alterar as informações registradas e contradição no endereço.

A solução proposta tem por objetivo automatizar a maior parte desse processo através do uso de tablet ou smartphone, munido de GPS, através da captura desses dados e cálculos matemáticos. Desse modo, o militar em serviço enviará diretamente para a aplicação todas as informações necessárias com apenas alguns toques na tela.

Este documento está dividido como segue: o capítulo 2 apresenta o referencial teórico, abordando a problemática como embasamento para o desenvolvimento deste trabalho. O capítulo 3 trata da metodologia utilizada e o material empregado. O capítulo 4 aborda o desenvolvimento da solução proposta, mencionando características, modelagem e requisitos do projeto. O capítulo 5 apresenta os testes realizados durante a construção do projeto. O capítulo 6 apresenta a avaliação final através dos resultados obtidos e por fim o capítulo 7 apresenta as referências bibliográficas.

## 2 REFERENCIAL TEÓRICO

### 2.1 APLICATIVOS WEB PROGRESSIVOS (PWA)

Os PWA representam uma nova abordagem para o desenvolvimento de aplicativos, no qual se faz uso das capacidades dos navegadores modernos para entregar ao usuário uma experiência similar à de aplicativos nativos<sup>3</sup> (Teixeira, 2016). Para o desenvolvimento do aplicativo faz-se uso de tecnologias geralmente utilizadas no desenvolvimento web, tais como Html, Css, Javascript dentre outras.

<sup>3</sup>São aplicativos criados especificamente para determinado Sistema Operacional.

---

Representam um avanço em relação às páginas web comuns pelo fato de acessarem recursos internos dos dispositivos, tais como GPS, acelerômetro, dentre outros e realizarem ações como lançar *push notifications*. Como limitações está o fato de não acessarem todos os recursos do dispositivo, ao contrário do que ocorre com os aplicativos nativos.

Para justificar o termo progressivo, o PWA funciona de modo que vai adquirindo funções à medida que o usuário se engaja com a página. A partir daí é sugerida a sua instalação e o PWA começa a se comportar como um aplicativo nativo, adicionando ícone na tela inicial do dispositivo, tendo processo próprio no Sistema Operacional e possibilitando funcionamento mesmo sem internet (LEPAGE, 2017).

## 2.2 REQUISITOS ESSENCIAIS DE UM PWA

Há três requisitos, conforme Malavolta (2016), para que um aplicativo *web mobile* seja considerado PWA:

O conteúdo deve ser provido através do protocolo HTTPS, a fim de garantir segurança ao aplicativo;

Possuir manifesto web do aplicativo<sup>4</sup> declarando informações do aplicativo como seu nome, ícones, URL de base;

Executar um *service worker*<sup>5</sup>.

## 2.3 VANTAGEM COMPETITIVA DO PWA

Existem muitas vantagens para se optar por um PWA, portanto serão abordadas nesse trabalho apenas três principais: custo para desenvolvimento, processo de obtenção de usuários e aprimoramentos.

Em relação aos custos e tempo para desenvolvimento, o PWA abre grande vantagem sobre os aplicativos Nativos. Atualmente, existem mais desenvolvedores web no mercado que desenvolvedores para as duas maiores plataformas atualmente: Java e Objective-C (RESHETILO; OPANASENKO, 2017) e isso faz com que automaticamente os custos para estes sejam maiores. Além disso, seriam necessárias duas equipes, uma para cada plataforma. A tabela 1 exemplifica este cenário com base nas contribuições dos autores supracitados.

---

<sup>4</sup> O manifesto dos aplicativos web é um arquivo JSON que permite controlar como o aplicativo web ou site é exibido para o usuário em áreas que normalmente se espera ver aplicativos nativos (por exemplo, a tela inicial de um dispositivo), como definir o que o usuário pode inicializar e o visual durante a inicialização (GAUNT; KINLAN, 2017).

<sup>5</sup> Um script de navegador executado em segundo plano, separado da página da Web, possibilitando recursos que não precisam de uma página da Web ou de interação do usuário (GAUNT, 2017).

**Tabela 1:** Comparação entre aplicativos nativos e PWA.

Aspecto	Aplicativos Nativos	Aplicativos Web Progressivo
Custo de Aquisição	Alto	Baixo
Tempo de desenvolvimento e custo	Alto	Baixo
Abrangência (entre dispositivos e plataformas)	Pequena	Larga
Performance	Rápida	Lenta
Acesso a funcionalidades específicas do dispositivo	Completo	Limitado
Processo de atualização	Lento	Rápido

Fonte: acervo pessoal

O Housing.com gasta \$3,75 para cada novo usuário de seu aplicativo *Android* e apenas 7 centavos para adquirir um usuário PWA (RESHETILO; OPANASENKO, 2017).

A segunda vantagem do PWA é o processo para obtenção de novos usuários. Para obter acesso a um PWA é necessário apenas um clique. Enquanto o usuário navega naturalmente pelo website, é sugerida a instalação. Ao aceitar, o processo de instalação é iniciado sem a necessidade de sair da página ou visitar uma loja de aplicativo e, em poucos segundos, o aplicativo está instalado com ícone acessível em sua tela inicial. Este mesmo processo se daria de forma bem menos envolvente em um aplicativo nativo. Primeiro, o usuário seria redirecionado para a loja de aplicativos em questão, aceitaria os termos e instalaria o aplicativo.

A terceira vantagem do PWA está no momento em que se faz necessário aprimorar o aplicativo, seja atualizando determinado recurso ou adicionando um novo. Nessa situação, todo o processo de desenvolvimento estará centralizado em um mesmo código, que afetará o website e o PWA em todas as plataformas que ele estiver sendo utilizado. Dessa forma, todos os usuários passam a receber uma versão atualizada, assim que estiverem online (RESHETILO; OPANASENKO, 2017).

## 2.4 DESVANTAGENS

A principal desvantagem do PWA é que ele está limitado à compatibilidade dos navegadores de internet existentes. Mas o ponto positivo é que, por se tratar de uma iniciativa fortemente apoiada pelo Google, o PWA vem ganhando rapidamente novos apoiadores.

No entanto, até mesmo nos navegadores que não dão suporte ao PWA, as páginas web serão servidas normalmente e a experiência apesar de simplificada será melhor que a de um site meramente responsivo. Estudos de caso mostram que os PWAs são melhores que os sites responsivos mesmo em navegadores não suportados. (RESHETILO; OPANASENKO, 2017).

Outra desvantagem está na velocidade dos PWAs. Estes ainda são mais lentos e consomem mais bateria que os aplicativos nativos, pois o dispositivo precisa interpretar o código de alto nível do PWA (RESHETILO; OPANASENKO, 2017).

---

### 3 METODOLOGIA

A metodologia utilizada neste trabalho seguiu as fases representadas abaixo:

#### 1ª. Análise do contexto atual de acompanhamento de viaturas do CBMTO;

#### 2ª. Levantamento de requisitos:

- a) Análise das necessidades dos usuários;
- b) Análise das formas de captura das informações;
- c) Análise do material necessário;
- d) Análise da viabilidade do uso de PWA.

#### 3ª. Prototipagem e desenvolvimento:

- a) Definição das tecnologias utilizadas;
- b) Definição das ferramentas utilizadas no desenvolvimento;
- c) Desenvolvimento da modelagem da aplicação;
- d) Integração da aplicação com o que já havia sido desenvolvido no CBMTO;
- e) Desenvolvimento da interface gráfica da aplicação;
- f) Análise de ferramentas adicionais caso necessário;
- g) Concepção de protótipo.

#### 4ª. Análises e Testes:

- a) Análise do funcionamento da aplicação;
- b) Análise da experiência do usuário;
- c) Teste em cenário real na viatura;
- d) Elaboração de relatório com base nos testes.

#### 5ª. Finalização do projeto:

- a) Aperfeiçoamento com base nos testes;
- b) Exposição dos dados obtidos.

### 3.1 MATERIAL UTILIZADO

Para a confecção da aplicação proposta foram utilizadas algumas ferramentas necessárias para o desenvolvimento da solução e dispositivos físicos para testes e simulações. Foi utilizado o PHPStorm 2018.1 como ambiente de desenvolvimento integrado. Como servidor da aplicação foi optado pelo WampServer 3.1.0, o qual conta com interpretador de Php na versão 7.1.9 e servidor de banco de dados Mysql na versão 5.7.19.

Em relação ao hardware, foram utilizados: um Notebook Asus N53S com processador Intel core i7, SSD Kingston de 240 GB e 8 GB de Memória Ram, executando o Windows 10 Pro 2018 como sistema

operacional; e um Smartphone Motorola, modelo Moto Z Play, de 2016.

## 4 DESENVOLVIMENTO DA APLICAÇÃO

O desenvolvimento da aplicação foi dividido em fases sistematicamente definidas, que estão descritas nos tópicos seguintes e foram obedecidas na ordem em que seguem.

### 4.1 LEVANTAMENTO DE REQUISITOS

Essa fase envolveu a análise do problema enfrentado pelo CBMTO relacionado ao registro das informações durante o deslocamento das viaturas para o atendimento às ocorrências. A partir daí, foi possível extrair a substância necessária para o entendimento do problema e desenvolvimento da solução mais adequada.

Considerando o fato de que essa aplicação será utilizada dentro de uma viatura de emergência, foram optados por dois princípios basilares:

- Somente informações fundamentais serão capturadas;
- A captura dessas informações, sempre que possível, será realizada de modo automatizado através dos recursos do dispositivo móvel em uso.

#### 4.1.1 Contextualização do problema

Atualmente, o acompanhamento ocorre da seguinte forma:

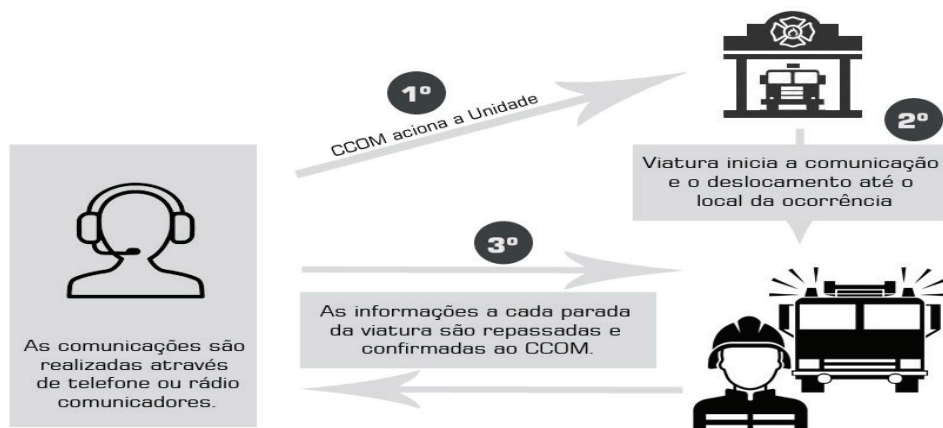
- A central de comunicações (CCOM) do CBMTO modula através de um rádio transmissor com a unidade que deverá atender a ocorrência;
- Passa as informações do incidente e o endereço;
- Os militares ao saírem da Unidade informam a central o horário e a quilometragem atual do veículo;
- Repetem o processo anterior ao chegarem no local do incidente;
- Repetem novamente ao chegarem no hospital ou outro local eventualmente necessário;
- Por fim, repetem ao retornarem à Unidade;
- Em todos esses passos, a CCOM insere esses dados em um sistema online.

Nesse cenário os principais problemas enfrentados são:

- A necessidade de tomar a atenção do motorista da viatura para a passagem da informação das quilometragens (odômetro atual da viatura);
- Disparidades em relação ao horário enviado e o registrado no sistema;
- Qualquer alteração que necessite ser feita em qualquer desses dados deve ser solicitada via rádio não podendo ser realizada pelo próprio militar que gerou a informação;
- Por não haver um método dedicado à captura do endereço do incidente, muitas vezes, a geração

de estatísticas fica prejudicada, em virtude das várias maneiras de se descrever o mesmo endereço. A Figura 1 exemplifica este cenário.

**Figura 1** - Comunicação do CBMTO durante deslocamento a ocorrências



Fonte: acervo pessoal

## 4.2 SOLUÇÃO PROPOSTA

Propõe-se o desenvolvimento de uma aplicação utilizável tanto através do navegador de internet, quanto como aplicativo instalável, em virtude das características do PWA, como mencionadas.

Como o projeto foi pensado no armazenamento apenas das informações fundamentais, a interface se apresentará bem limpa e o processo será simplificado.

Considerando o contexto em questão, as informações a serem guardadas serão:

- Dados da viatura que atendeu a ocorrências;
- Dados dos militares que estavam na viatura;
- Dados da ocorrência;
  - Natureza;
  - Endereço da Ocorrência.
- Dados do Atendimento;
  - Horário e coordenadas geográficas do acionamento da viatura;
  - Horário e coordenadas geográficas na chegada ao local da ocorrência;
  - Horário de saída do local da ocorrência;
  - Horário e coordenadas geográficas no despacho do atendido (Hospital, e outros);
  - Horário de saída do local de despacho;
  - Horário de entrada na base (Unidade);



- Se outra ocorrência surgir no deslocamento da saída do despacho até a Unidade, serão armazenados o horário e a coordenada deste ponto.

A interface gráfica será minimalista, com botões grandes e textos bem visíveis. Todas as informações, exceto as coordenadas geográficas, poderão ser alteradas pelos próprios componentes em serviço na viatura, evitando que seja necessária a intervenção de terceiros e a passagem de informações via rádio ou telefone, como ocorre atualmente no CBMTO.

#### 4.2.1 Sistema Legado

Muitas informações necessárias para o funcionamento adequado da aplicação proposta serão capturados antecipadamente através do sistema de Intranet, já em uso no CBMTO.

#### 4.2.2 Contextualização do sistema de Intranet

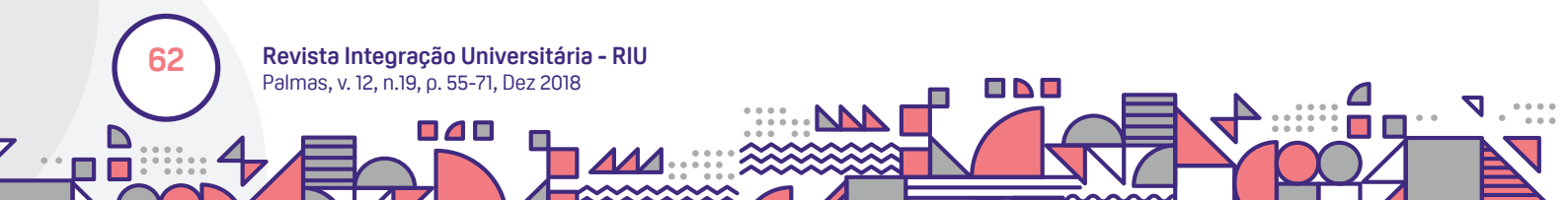
Para que uma viatura preste um atendimento no CBMTO é necessário que ela esteja registrada no sistema, bem como estar vinculada a um determinado dia de serviço, o que nesse contexto é chamado de Escala de Serviço Operacional.

Nestas escalas ficam registrados as viaturas que irão operar naquele dia, bem como os militares que estarão empregados nelas. Também são registradas as funções de cada militar em serviço.

Essas informações serão imprescindíveis para o funcionamento da aplicação proposta, pois através delas será possível vincular uma viatura e seus militares a uma ocorrência atendida. Desse modo, ao final do dia, será possível obter o relatório de todos os atendimentos de todas as viaturas em serviço para aquele dia, acrescido das informações capturadas durante cada deslocamento, que são as quilometragens, horários, e coordenadas geográficas de cada ponto onde a viatura parou. A captura das informações dos deslocamentos dessas viaturas representa o objetivo da aplicação proposta.

### 4.3 MODELAGEM

A Figura 2 representa uma versão simplificada do diagrama de classes da aplicação proposta, onde é possível ter uma visão geral de como as informações serão organizadas.







- *background\_color*: Código html em hexadecimal para a cor que será usada no fundo da tela de abertura;

- *icons*: Recebe três parâmetros:

- *src*: endereço da imagem para a tela de abertura;

- *sizes*: dimensão da imagem;

- *type*: tipo da imagem.

Existem outras informações que podem ser passadas através do manifesto, como por exemplo a orientação de tela, mas para o projeto em questão estas mencionadas são suficientes.

**Figura 3** - representa o arquivo *manifest.json* da aplicação proposta.

```
manifest.json x
1 {
2   "short_name": "SIOCB",
3   "name": "Sistema de Operações do Corpo de Bombeiros",
4   "lang": "pt-BR",
5   "start_url": "/",
6   "display": "standalone",
7   "theme_color": "#6E0909",
8   "background_color": "#ffffff",
9   "icons": [
10    {
11      "src": "/files/img/icon.png",
12      "sizes": "512x512",
13      "type": "image/png"
14    }
15  ]
16 }
```

Fonte: acervo pessoal

Um *service worker* é um script executado em segundo plano pelo navegador, que viabiliza acesso a recursos antes possíveis apenas para aplicativos nativos (GAUNT, 2017).

Antes dos *service worker*, as aplicações web tinham pouca interação com os recursos de hardware do usuário, estando limitadas a essencialmente exibirem informações aos usuários através do navegador. Tarefas como receber entrada de dados pelo microfone ou *webcam* já eram possíveis, mas capturar coordenadas através do GPS do *smartphone* eram inconcebíveis. Os *service workers* tornaram possível a comunicação das aplicações web com uma série de recursos de hardware presente na maioria dos dispositivos móveis, tais como GPS, acelerômetro, câmeras, dentre outros.

#### 4.4.2 Interface gráfica

A aplicação proposta, apesar de poder ser usada tanto em modo retrato como paisagem, foi desenhada priorizando-se o modo paisagem, em vista do contexto de utilização (tablet no painel da viatura). Por isso, as imagens apresentadas neste documento estão em modo paisagem. A tela de abertura do aplicativo (*splashscreen*), definida através do arquivo *manifest.json*, está representada na Figura 4.

Figura 4 – Tela de abertura do aplicativo



Fonte: acervo pessoal

Depois do *splashscreen* o usuário é direcionado a página de *login* (Figura 5), para a sua autenticação e, em seguida, entrar no aplicativo.

Figura 5 – Tela de login do aplicativo

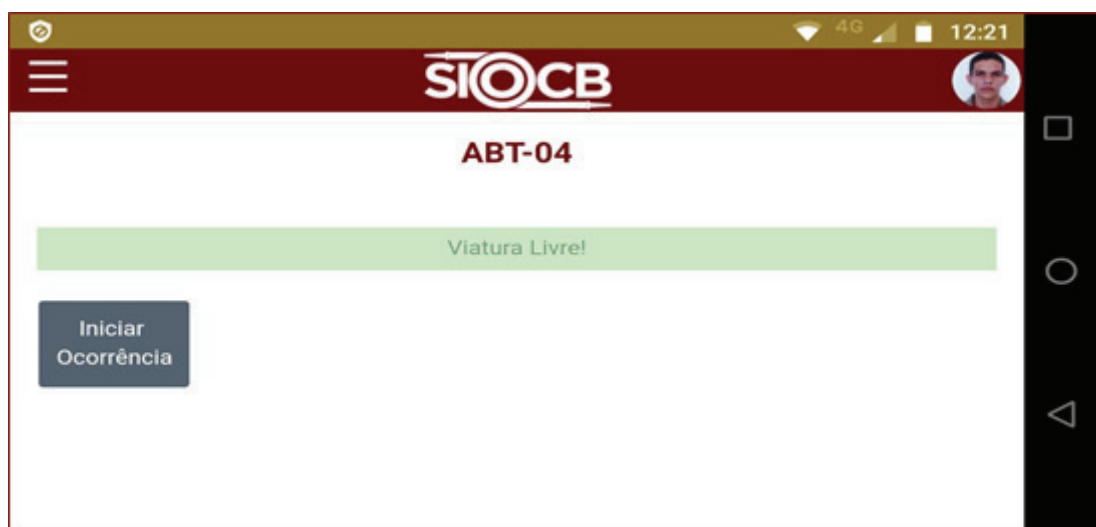


Fonte: acervo pessoal

Ao realizar o *login* e entrar no sistema, o usuário é direcionado a página inicial (Figura 6), onde é exibido a viatura na qual ele está de serviço.

Como fora definido no projeto, se optará por telas limpas com apenas informações fundamentais. Portanto, na tela inicial (Figura 6), é exibido apenas o nome da viatura (ABT-04), status (Viatura Livre!), e um botão para iniciar uma nova ocorrência.

Figura 6 – Tela inicial



Fonte: acervo pessoal

O uso da aplicação se inicia a partir do momento em que são recebidas as instruções para atendimento a ocorrência: natureza da Ocorrência, características e Endereço. Ao entrarem na viatura, os militares devem iniciar o processo.

Ao tocar no botão “Iniciar Ocorrência”, o aplicativo captura a coordenada geográfica e o horário corrente, conforme Figura 7.

Figura 7 – Tela do aplicativo depois de iniciado o deslocamento



Fonte: acervo pessoal

Ao chegarem no local da ocorrência, o botão ‘Viatura no Local’ deve ser tocado. Dessa forma, as coordenadas geográficas do local serão capturadas. Nesse momento duas consultas são realizadas: uma consulta a *api* do *Google Maps* responsável por converter coordenadas geográficas em endereços, e uma consulta a outra *api* no *Google Maps*, responsável por calcular a distância entre uma coordenada e outra.

Capturado o endereço do local, este é inserido no banco de dados. Também é realizado

o cálculo que incrementa o valor do hodômetro, somando-se o valor anterior à distância percorrida pelo veículo. Em seguida, são apresentados na tela: o endereço capturado, com botão para editar e/ou adicionar número do lote e complemento, o status atualizado e um ponto a *timeline* representando a viatura no local da ocorrência. Abaixo deste ponto, são apresentados o horário e o km naquele instante, conforme Figura 8. Também é possível alterar todas as informações capturadas até esse ponto, exceto as coordenadas geográficas.

**Figura 8** – Tela do aplicativo no local da ocorrência



Fonte: acervo pessoal

Uma vez no local, o atendimento padrão é realizado. Esse contexto pode resultar em duas situações: o atendimento ser realizado e concluído no local da ocorrência, ou fazer-se necessário o deslocamento para um segundo local, o que na maioria dos casos pode ser um hospital, no caso de atendimento a pessoas, ou despacho de animais para algum órgão ou sua devolução à natureza.

No primeiro caso, o botão finalizar ocorrência pode ser pressionado, então será registrado que esta ocorrência terminou no próprio local de atendimento. Será armazenado os horários e hodômetro atualizado e a viatura iniciará seu deslocamento até a Unidade. No segundo caso, o botão 'Saída para Despacho' deve ser pressionado, então serão exibidas algumas opções para se especificar qual o tipo de despacho, conforme Figura 9.

**Figura 9** – Opções de saída para despacho



Fonte: acervo pessoal

Ao tocar sobre o botão adequado para o tipo de despacho, são registrados o horário de saída do local e o status é alterado, informando qual o tipo de despacho em andamento, e o horário em que este foi iniciado (Figura 10). Também é substituído o botão “Saída para Despacho” pelo botão “Entrada no Despacho”.

**Figura 10** – Exibição do Status conforme tipo de despacho escolhido



Fonte: acervo pessoal

Chegando ao local de despacho, o botão Entrada no Despacho é pressionado. Assim, são capturadas as coordenadas geográficas do local, horário corrente e é realizado o cálculo para atualização do hodômetro. Feito isso, os dados são apresentados na tela, o botão “Entrada no Despacho” é removido e o status é atualizado, conforme Figura 11.

**Figura 11** – Tela do aplicativo na entrada do despacho



Fonte: acervo pessoal

Ao terminar o despacho da ocorrência, o botão “Finalizar Ocorrência” deve ser tocado. Ao fazê-lo, serão exibidos botões para especificar a próxima atividade da viatura (Figura 11). Selecionado o botão adequado, a aplicação armazena o horário corrente e atualiza o status da ocorrência (Figura 12).

Ao retornar à Unidade, o botão Viatura no PZ (Figura 12) deve ser pressionado. Desse

modo, o horário corrente será armazenado, o cálculo da distância será feito entre a última coordenada capturada e a coordenada da unidade, o hodômetro será atualizado e a viatura estará livre novamente para atender a uma nova ocorrência e repetir o ciclo aqui descrito.

**Figura 12** – Tela do aplicativo no retorno a Unidade



Fonte: acervo pessoal

## 5 TESTES

Para avaliar a aplicação proposta foram destacados três fatores principais: a compatibilidade, a usabilidade e a eficiência. O primeiro para definir se a aplicação se enquadraria como um *Progressive Web App*, o segundo para avaliar a qualidade da interação do usuário com a aplicação, e o terceiro para medir o quão solucionado estaria o problema objeto deste artigo.

Para realizar os testes, a aplicação proposta foi hospedada no servidor de aplicação do CBMTO.

### 5.1 TESTE DE COMPATIBILIDADE

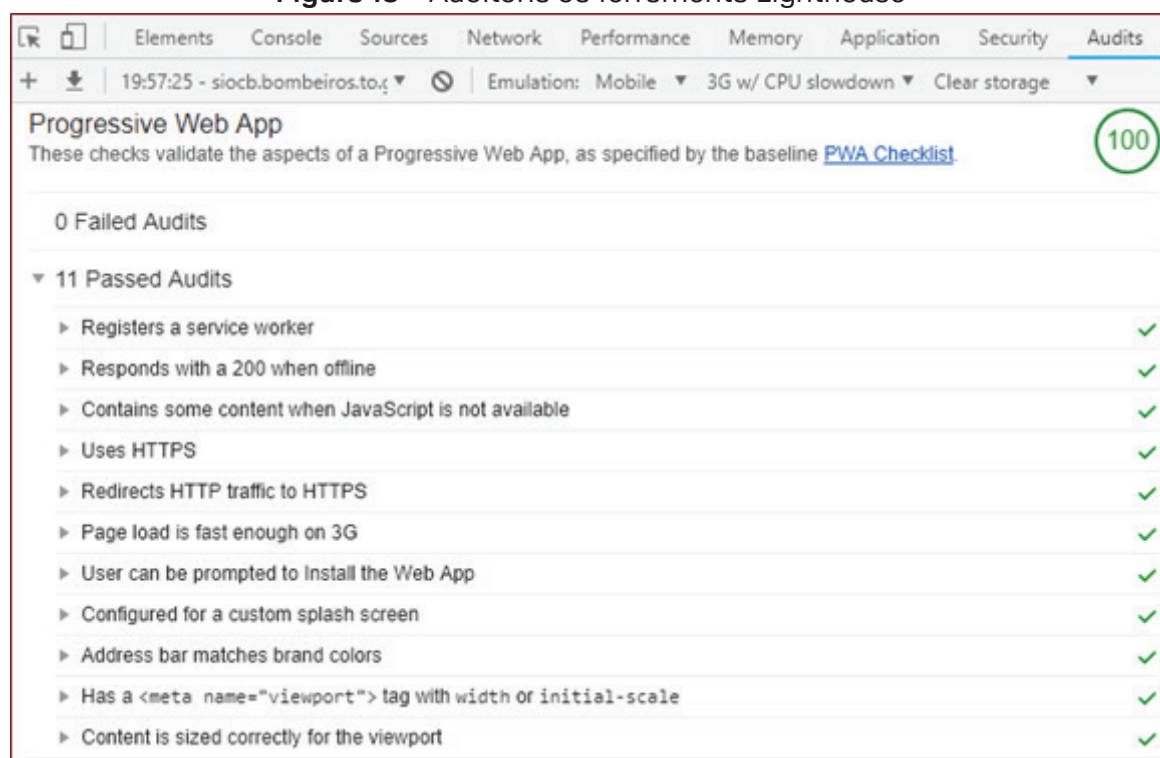
Como parte dos esforços do Google para garantir a melhor experiência possível para os usuários da internet, foi criada uma ferramenta denominada *Lighthouse*<sup>8</sup>, a qual realiza uma análise capaz de avaliar cinco características de qualquer página da web, que são: performance, melhores práticas, acessibilidade, SEO e compatibilidade com PWA. A análise realizada considerou apenas o último aspecto.

Existem várias formas de se utilizar o *Lighthouse* segundo sua documentação<sup>9</sup> oficial, porém utilizamos a extensão para o navegador. Segundo a ferramenta, a aplicação proposta passou por 11 auditorias, nas quais teve 100% de aprovação, conforme Figura 13.

<sup>8</sup> O *Lighthouse* é uma ferramenta automatizada e de código aberto para melhorar a qualidade das páginas da web.

<sup>9</sup> A documentação oficial do *Lighthouse* está disponível em: <https://developers.google.com/web/tools/lighthouse/>.

Figura 13 – Auditoria da ferramenta Lighthouse



Fonte: acervo pessoal

## 5.2 TESTE DE USABILIDADE

Para analisar a qualidade da interface gráfica da aplicação proposta, ela foi utilizada em situações reais por militares em serviço. Os principais pontos abordados foram o tamanho e posição dos botões, e a facilidade para editar os campos.

No primeiro ponto, a aplicação teve 100% de aprovação, pelo tamanho, cor e posição adequada dos botões.

No segundo ponto, a aplicação não atingiu a unanimidade de aprovação pela dificuldade natural que se tem de digitar texto em uma viatura em movimento. Nesse ponto, é desejável que se planeje um método de inserção de texto facilitado na aplicação, o que pode ser desenvolvido em trabalhos futuros.

## 5.3 TESTE DE EFICIÊNCIA

Essa fase envolveu uma gama de testes com a aplicação a fim de determinar os níveis de estabilidade e desempenho desta antes de sua implantação. Nessa fase concluiu-se que:

A aplicação necessita de internet para ter pleno funcionamento;

A aplicação capturou corretamente as coordenadas em todos os testes;

O cálculo da distância apresentou disparidades em alguns casos durante os testes.

Como a aplicação calcula a distância considerando o percurso mais curto entre duas coordenadas,



---

pode ocorrer de a viatura realizar outro percurso, acarretando a diferença entre o valor calculado e o valor da distância real.

Considerando os pontos levantados, a aplicação apresentou bom nível de eficiência e com alguns ajustes será capaz de realizar a tarefa para a qual foi desenvolvida em sua plenitude.

## 6 ANÁLISE FINAL E AVALIAÇÃO

Considerando-se todos os testes realizados, os recursos utilizados, o tempo e esforços dedicados ao desenvolvimento desta solução, a mesma apresentou grande potencial de ser aplicada no registro de deslocamentos do CBMTO de forma real e definitiva.

Uma ressalva para a captura de coordenadas geográficas, onde se notou que seria adequado capturá-las durante todo o trajeto, não se limitando aos pontos onde a viatura parar, mas em curtos intervalos de tempo para ser possível montar o trajeto real pelo qual a viatura passou, bem como calcular a distância exata dos deslocamentos.

Outra melhoria poderia ser realizada aumentando as fontes das telas para edições de endereço, hodômetros e horários, melhorando a experiência do usuário (militar em serviço). Realizado essas melhorias, a aplicação estará em condições de ser colocada para uso real em ambiente de produção.

## REFERÊNCIAS

ATER, T. **Building Progressive Web Apps**. Sebastopol: O'Reilly Media Inc, 2017.

COMSCORE (Estados Unidos) (Comp.). **Relatório sobre dispositivos móveis**. 2016. Disponível em: <<https://www.comscore.com/>>. Acesso em: 20 nov. 2017.

GAUNT, M. **“Service Workers: an Introduction”**. Disponível em: <https://developers.google.com/web/fundamentals/primers/service-workers>>. Acesso em: 10 nov. 2017.

LEPAGE, P. **“Your First Progressive Web App”**. 2017. Disponível em: <<https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp>>. Acesso em: 12 nov. 2017.

MALAVOLTA, I. **Beyond Native Apps: Web Technologies to the Rescue!** Amsterdam: Proceedings of the 1st International Workshop on Mobile Development, 2016.

RESHETILO, K; OPANASENKO, S. **“Progressive Web Apps vs Native: Which Is Better for Your Business?”** 2017. Disponível em: <<https://www.technology.org/2017/07/28/progressive-web-apps-vs-native-which-is-better-for-your-business>>. Acesso em: 11 nov. 2017.

STERLING, G. **“Mobile now accounts for nearly 70% of digital media time [comScore]”**. 2017. Disponível em: <<https://marketingland.com/mobile-now-accounts-nearly-70-digital-media-time-comscore-210094>>. Acesso em: 10 nov. 2017.

TEIXEIRA, F. **“O que são Progressive Web Apps?”** 2016. Disponível em: <https://brasil.uxdesign.cc/o-que-s%C3%A3o-progressive-web-apps-86e1b5306051>. Acesso em: 05 fev. 2018.