

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA CURSO DE  
ESPECIALIZAÇÃO EM TECNOLOGIA JAVA

CARLOS HUMBERTO LOPES COSTA

**DESENVOLVIMENTO DE APLICATIVO MÓVEL DO SERVIÇO WEB  
PATROOL.COM UTILIZANDO A TECNOLOGIA APPCELERATOR TITANIUM**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA – PARANÁ

2012

CARLOS HUMBERTO LOPES COSTA

**DESENVOLVIMENTO DE APLICATIVO MÓVEL DO SERVIÇO WEB  
PATROOL.COM UTILIZANDO A TECNOLOGIA APPCELERATOR TITANIUM**

Monografia de Especialização apresentada ao Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Especialista em Tecnologia Java”.

Orientador: Prof. Nelson Kashima

CURITIBA – PARANÁ

2012

## **AGRADECIMENTOS**

À minha noiva Lorena Nadolny pela paciência, compreensão e incentivo que tornaram possível a conclusão desta monografia.

Aos meus pais que, com muito carinho e apoio, não mediram esforços para me apoiar em todas as etapas de minha vida e me fornecer a melhor educação.

## RESUMO

Este trabalho apresenta o desenvolvimento de um aplicativo *mobile* do serviço web intitulado “PATROOL.com” utilizando uma tecnologia alternativa a programação nativa em Java/Android. Serão apresentadas tecnologias como Appcelerator Titanium, Geolocalização e Google Maps API. A utilização da plataforma Appcelerator Titanium permitirá que o aplicativo seja desenvolvido uma única vez e disponibilizado para várias plataformas como: iOS, Android e BlackBerry.

**Palavras-chave:** Mobile, Android, Appcelerator Titanium, JavaScript.

## **ABSTRACT**

This work presents the development of a mobile application from a web service called "PATROOL.com" using an alternative technology to native programming in Java/Android. Technologies will be presented as Appcelerator Titanium, Geolocation and Google Maps API. The use of the platform Appcelerator Titanium will allow the application to develop once and available for various platforms such as: iOS, Android and BlackBerry.

**Keywords:** Mobile, Android, Appcelerator Titanium, JavaScript.

## LISTA DE FIGURAS

Figura 1. Diagrama de Casos de Uso.....	18
Figura 2. Diagrama de Sequência - Buscar Estatísticas de Problemas.....	20
Figura 3. Diagrama de Sequência - Relatar Problemas.....	20
Figura 4. Configurando a IDE Titanium Studio.....	21
Figura 5. Instalando as SDKs através da IDE Titanium Studio.....	22
Figura 6. Trecho do código-fonte mostrando como se obtém dados remotos.....	23
Figura 7. SQL que contabiliza os problemas próximos a localidade do usuário.....	24
Figura 8. Localização do usuário e as estatísticas de problemas da região.....	25
Figura 9. Aplicativo exibindo o relato de um problema. ....	25
Figura 10. Inserção da descrição do problema. ....	25
Figura 11. Visualização de índices de relatos de determinada região.....	25
Figura 12. Visualização de relatos em lista.....	27
Figura 13. Visualização de relatos em mapa. ....	27
Figura 14. Visualização de detalhes de relato. ....	27

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
GPS	Global Positioning System
IDE	Integrated Development Environment
SDK	Software Development Kit
SQL	Structured Query Language
SUSEP	Superintendência de Seguros Privados
UI	User Interface
UML	Unified Modeling Language
UNODC	United Nations Office on Drugs and Crime
URL	Uniform Resource Locator

## SUMARIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>10</b>
1.1	OBJETIVO .....	10
1.2	JUSTIFICATIVA .....	11
1.3	OPORTUNIDADES .....	11
1.4	ESCOPO .....	11
1.5	ORGANIZAÇÃO DO DOCUMENTO .....	12
<b>2</b>	<b>ESTUDO DAS TECNOLOGIAS.....</b>	<b>13</b>
2.1	O SERVIÇO WEB PATROOL.....	13
2.2	APPCCELERATOR TITANIUM.....	13
2.2.1	<i>Titanium Mobile SDK.....</i>	<i>13</i>
2.2.2	<i>Titanium JavaScript Enviroment.....</i>	<i>13</i>
2.2.3	<i>Titanium JavaScript API.....</i>	<i>14</i>
2.2.4	<i>Funcionamento da Plataforma Titanium.....</i>	<i>14</i>
2.2.5	<i>Vantagens e Desvantagens .....</i>	<i>14</i>
2.3	GEOLOCALIZAÇÃO .....	15
2.4	GOOGLE MAPS .....	16
2.4.1	<i>Google Maps API .....</i>	<i>16</i>
<b>3</b>	<b>DESENVOLVIMENTO .....</b>	<b>17</b>
3.1	ESPECIFICAÇÃO DE REQUISITOS .....	17
3.1.1	<i>Requisitos Funcionais .....</i>	<i>17</i>
3.1.2	<i>Requisitos Não Funcionais.....</i>	<i>17</i>
3.2	MODELAGEM DO SISTEMA.....	17
3.2.1	<i>Diagrama de Casos de Uso .....</i>	<i>18</i>
3.2.2	<i>Diagrama de Sequência.....</i>	<i>20</i>
3.3	IMPLEMENTAÇÃO .....	21
3.3.1	<i>Instalando o Titanium Studio.....</i>	<i>21</i>
3.3.2	<i>Configurando o Titanium Studio.....</i>	<i>21</i>
3.4	DESENVOLVENDO O APLICATIVO .....	22
3.4.1	<i>Utilizando a Linguagem JavaScript .....</i>	<i>22</i>
3.4.2	<i>Obtendo dados Remotamente .....</i>	<i>23</i>
3.4.3	<i>Obtendo estatísticas da localização .....</i>	<i>23</i>
3.5	PUBLICANDO UM PROBLEMA NO SERVIÇO PATROOL.....	24
3.6	APRESENTAÇÃO DO APLICATIVO MOBILE .....	24
<b>4</b>	<b>CONCLUSÃO .....</b>	<b>26</b>



4.1 TRABALHOS FUTUROS .....26

**5 REFERÊNCIAS.....28**

# 1 INTRODUÇÃO

O PATROOL surgiu como inspiração da conhecida frase de Dalai Lama - *"Seja a mudança que você quer ver no Mundo"*. Acredita-se que através de pequenas iniciativas é possível gerar grandes mudanças no Mundo.

É um jeito diferente de utilizar a tecnologia a favor da sociedade. É exercer a cidadania de forma prática, eficiente e frequente. É também a oportunidade da mudança através da voz, de uma sociedade que já não se conforma com as injustiças, com o descaso, com a insegurança e clama por melhor qualidade de vida.

PATROOL.com acredita na voz, como um meio para solucionar os problemas da sociedade. Desta forma, PATROOL.com oferece um meio para relatar problemas do nosso cotidiano e propor ideias de melhoria para esses problemas. Pode-se também utilizá-lo para se manter informado sobre tudo o que acontece nos locais de seu interesse.

Comunicar o extravio de documentos, denunciar vandalismo, propor melhorias em um cruzamento com alto índice de acidentes e solicitar mais iluminação para uma rua perigosa são outros exemplos de utilização do PATROOL.com.

O acesso via smartphone, objetivo principal desse trabalho, permite por exemplo, consultar se uma rua é segura ou não para estacionar o carro. Usuários mais distraídos também poderão configurar alertas específicos de acordo com sua localização. Futuramente, por meio de uma SmartTV será possível consultar no PATROOL se há acidentes no trajeto de casa ao escritório.

Não se trata de mais uma rede social, ao contrário disso, PATROOL.com estará integrado as principais redes sociais de modo a entregar uma experiência completa aos usuários da plataforma.

## 1.1 Objetivo

O objetivo do presente trabalho é cruzar dados de localização do dispositivo móvel do usuário, com dados do serviço PATROOL.com, permitindo ao usuário, por exemplo, identificar se um local é seguro para estacionar um veículo.

## **1.2 Justificativa**

Vivemos um momento delicado em que as estatísticas mostram uma sociedade diferente daquela com a qual sonhamos.

O Escritório das Nações Unidas para Drogas e Crimes (UNODC) recentemente apresentou um relatório indicando que em todo o mundo ocorreram 468 mil homicídios oficiais em 2010. Sendo o Brasil responsável pelo maior número absoluto de homicídios do mundo.

Porém as estatísticas de criminalidade baseiam-se apenas em dados oficiais, desta forma, nem sempre essas estatísticas correspondem a realidade. Segundo pesquisas, apenas um em cada três crimes ocorridos no Brasil são registrados nos órgãos competentes. Em São Paulo, 68% das vítimas de crimes não relatam oficialmente o caso, já no Paraná esse número sobe para 75%. As vítimas alegam não acreditarem na investigação policial, não querem se expor ou têm medo de represálias.

A utilização de dispositivos móveis aliados à computação na nuvem e a geolocalização, vem revolucionando a forma como as pessoas interagem com a informação. O objetivo é aproveitar todo o potencial e benefícios da mobilidade para fornecer mais um canal de interação a plataforma PATROOL. Um canal que dará voz a uma sociedade que clama por mais segurança e qualidade de vida.

## **1.3 Oportunidades**

Em decorrência da insegurança e da violência, o mercado de segurança pública e privada, apenas no Brasil, movimentou aproximadamente R\$ 60,6 bilhões em 2010. Com a proximidade da Copa do Mundo e as Olimpíadas, serão injetados mais R\$ 717 milhões em segurança até 2013. Somente o segmento de segurança eletrônica movimentou R\$ 3,2 bilhões em 2011 no Brasil, com crescimento de 12% com relação ao ano anterior. Segundo dados da SUSESP, no ano de 2010 somente as seguradoras brasileiras desembolsaram R\$ 12 bilhões em indenizações, apenas com sinistro de automóveis.

## **1.4 Escopo**

Esse projeto se restringe ao desenvolvimento de um aplicativo móvel protótipo para a plataforma Android 2.2 (API 8), o qual contará com a funcionalidade

específica de visualização de índices de problemas de uma determinada região e a possibilidade de o usuário relatar seu próprio problema. Esse aplicativo deverá ser capaz de buscar a localização do usuário e exibir as estatísticas de problemas na região em que o usuário estiver.

### **1.5 Organização do documento**

O presente trabalho está organizado em três capítulos intitulados respectivamente como: fundamentação teórica, desenvolvimento e conclusões. O capítulo 2 apresenta os aspectos teóricos estudados para o desenvolvimento do trabalho. No capítulo 3 é descrito como foi realizado o desenvolvimento deste trabalho, detalhando os requisitos do protótipo, a especificação e a implementação. Por fim, o capítulo 4 traz conclusões deste trabalho, bem como alguns aspectos que ficaram em aberto, servindo de sugestões para futuras implementações.

## **2 ESTUDO DAS TECNOLOGIAS**

Nesse capítulo são apresentadas os serviços e as tecnologias utilizadas nesse trabalho.

### **2.1 O serviço web PATROOL**

PATROOL.com é uma plataforma web onde é possível acompanhar o que acontece em um bairro e conectar-se com a vizinhança. É muito fácil utilizar o PATROOL, basta criar uma conta, ou utilizar sua conta do Facebook, Twitter, Google ou Yahoo, e então escolher quais lugares você gostaria de monitorar – por exemplo, sua casa e a escola dos filhos. Desta forma, o PATROOL exibirá informações personalizadas e atualizadas sobre tudo o que está acontecendo perto de seus lugares prediletos, e também o que as pessoas estão falando nesses lugares.

É um local para relatar problemas em sua rua, juntar-se a conversas propondo ideias de melhorias para um bairro, fazer perguntas a seus vizinhos, informar eventos em sua localização, ou apenas se manter informado sobre tudo o que está acontecendo naquela região. É possível também obter um e-mail diário com todas as notícias de seus lugares prediletos.

### **2.2 Appcelerator Titanium**

Appcelerator Titanium é uma plataforma de desenvolvimento de aplicativos que permite criar aplicações móveis utilizando a simplicidade da linguagem JavaScript. Os aplicativos desenvolvidos através do Appcelerator Titanium são executados como aplicativos nativos em plataforma iOS, Android, BlackBerry, entre outras.

#### **2.2.1 Titanium Mobile SDK**

O Titanium Mobile SDK permite criar, executar e empacotar aplicativos móveis nativos para iOS, Android, e dispositivos BlackBerry utilizando sua JavaScript API. Aplicativos móveis Titanium são executados em um motor JavaScript autônomo que chama APIs nativas.

#### **2.2.2 Titanium JavaScript Environment**

Em uma aplicação Titanium o código-fonte é empacotado em um arquivo binário e, em seguida, interpretado em tempo de execução por um motor JavaScript embutido pelo Titanium no processo de *building*.

Titanium executa um aplicativo JavaScript utilizando um dos seguintes interpretadores: *JavaScriptCore* no iOS (o interpretador usado por Webkit) e *Rhino Mozilla* no Android e BlackBerry.

### 2.2.3 Titanium JavaScript API

A Titanium JavaScript API dá acesso a centenas de componentes UI nativos e componentes não-visuais dentro de uma aplicação. Alguns exemplos de componentes dessa APIs são o Titanium.UI (para User Interface) e Titanium.Network (para redes).

### 2.2.4 Funcionamento da Plataforma Titanium

A plataforma Titanium funciona da seguinte maneira, o código escrito em JavaScript é combinado com a API do Titanium (que é escrito na linguagem nativa do dispositivo de destino), que então é avaliado em tempo de execução por um interpretador JavaScript do próprio sistema operacional do dispositivo.

Quando um aplicativo Titanium é iniciado, uma execução JavaScript é criada na plataforma nativa e o código JavaScript é injetado como um objeto embutido em tempo de execução. A API Titanium funciona como uma ponte, expondo o acesso direto aos componentes e recursos próprios de cada plataforma, proporcionando assim uma experiência de aplicativo nativo.

É importante ter a clara compreensão sobre a estrutura do Titanium. A principal, é que o Titanium não utiliza um componente web de exibição como outras ferramentas, por exemplo o PhoneGap ([phonegap.com](http://phonegap.com)). O código JavaScript utilizado pelo Titanium não é *cross-compilado* nas respectivas línguas nativas. O código JavaScript é avaliado em tempo de execução. Titanium fornece acesso aos controles nativos de interface do usuário e animações, em vez de replicar isso através de CSS e JavaScript.

### 2.2.5 Vantagens e Desvantagens

Uma das maiores vantagens da plataforma Titanium é que ela dá um acesso de nível mais alto as APIs nativas de cada tipo de dispositivo. Titanium oferece interface e componentes, que são os mesmos que estão disponíveis nativamente tanto para Android como para iOS, que o diferencia de outras abordagens de desenvolvimento híbrido.

É possível também estender a API do Titanium para atender às necessidades específicas de uma aplicação. Outro grande benefício é que o Titanium possibilita aos desenvolvedores utilizarem seus conhecimentos em JavaScript, uma linguagem bastante comum, ao invés de ter que aprender a linguagem específica de cada plataforma. Finalmente, depois de ter concluído uma aplicação, Titanium oferece uma forma automática para empacotar e distribuir as aplicações para as lojas de aplicativos de cada plataforma através de um IDE chamado Studio Titanium.

Uma das desvantagens de se utilizar o Titanium como framework é que ele ainda não oferece acesso a todos os recursos do dispositivo. Isto se deve principalmente ao fato de que a API do Titanium conecta-se diretamente a plataforma nativa, e dessa forma as atualizações nas plataformas levam um tempo para estarem disponíveis na API do Titanium. Por esta razão, Titanium suporta apenas as plataformas mais recentes do Android, iOS, e alguns recursos em navegadores web.

Titanium nunca poderá apoiar todas as APIs nativas e funcionalidades dos dispositivos, que continuam a crescer e se desenvolver nos sistemas operacionais nativos. Mas Titanium se esforça para oferecer pelo menos 90 por cento dessas APIs e funcionalidades nativas nas plataformas suportadas.

### **2.3 Geolocalização**

Os celulares, assim como como outros dispositivos móveis, se tornaram bastante poderosos ao adotarem tecnologias que permitem identificar a localização do aparelho. Algumas das tecnologias utilizadas por esses dispositivos que permitem identificar a localização são a triangulação de torres de celulares e o posicionamento por satélite via GPS - *Global Positioning System*.

Por meio da combinação do posicionamento geográfico com dados de fontes variadas pode-se criar aplicações inteligentes, como é o caso do aplicativo fruto desse trabalho, que tem como objetivo cruzar dados da localização do dispositivo com dados do serviço PATROOL.com, permitindo ao usuário, por exemplo, identificar se o local é seguro para estacionar um veículo.

## 2.4 Google Maps

É um serviço gratuito para visualização de mapas desenvolvido pela empresa Google. Ele oferece mapas de ruas, um planejador de rotas para se locomover de carro, a pé, bicicleta ou transporte público. A funcionalidade que mais chama atenção é o *Google Street View*, que permite explorar lugares no mundo todo através de imagens em 360 graus da rua. É possível por exemplo dar uma olhada em restaurantes, visitar bairros e até mesmo parques.

O *Google Maps* alimenta também outros serviços da Google baseados em mapas como *Google Street View*, *Google Transit*, *Google Moon* e até mesmo serviços de terceiros através da *Google Maps API*.

### 2.4.1 Google Maps API

Em Junho de 2005 a empresa Google lançou o *Google Maps API*, um serviço gratuito que permite a qualquer um incorporar um mapa do Google em uma página web ou aplicativo usando JavaScript, permitindo ainda a manipulação desse mapa e a inclusão de conteúdos/marcações.

O *Google Maps API* é gratuito para uso comercial desde que o aplicativo que esteja utilizando o serviço seja acessível ao público, esse acesso não seja cobrado e não esteja gerando mais de 25.000 acessos a mapas por dia. O aplicativo que não atender a esses requisitos pode comprar o *Google Maps API Premier*.



### 3 DESENVOLVIMENTO

As seções seguintes descrevem a especificação, implementação e a operacionalidade da aplicação mobile.

#### 3.1 Especificação de Requisitos

Na sequência são apresentados os Requisitos Funcionais e Requisitos Não Funcionais que são atendidos pela aplicação.

##### 3.1.1 Requisitos Funcionais

- O aplicativo mobile deverá obter remotamente as estatísticas de problemas identificadas próximo a localização do dispositivo, em um raio de 03 quilômetros.
- O aplicativo deverá exibir um mapa apontando a localização do dispositivo.
- O aplicativo deverá permitir ao usuário relatar um problema diretamente no serviço web PATROOL.com.

##### 3.1.2 Requisitos Não Funcionais

- O aplicativo mobile deve ser acessível por dispositivos móveis Android versão 2.2 (API 8) ou mais recente.
- A curva de aprendizagem da tecnologia a ser empregada deverá ser reduzida em comparação a outras tecnologias;
- Boa performance da aplicação.
- O aplicativo necessitará de conexão com a internet.
- O aplicativo deverá obter a localização geográfica do usuário utilizando o dispositivo GPS ou a rede de comunicação.

#### 3.2 Modelagem do Sistema

Na sequência é apresentada a especificação do aplicativo, onde foram utilizados conceitos da orientação a objetos e a Unified Modeling Language (UML) para a criação do diagrama de casos de uso e de sequência. Para a construção dos diagramas foi utilizado a ferramenta online *Creately* ([creately.com](http://creately.com)).

### 3.2.1 Diagrama de Casos de Uso

O aplicativo mobile possui dois cenários de caso de uso que são ilustrados abaixo na Figura 1.

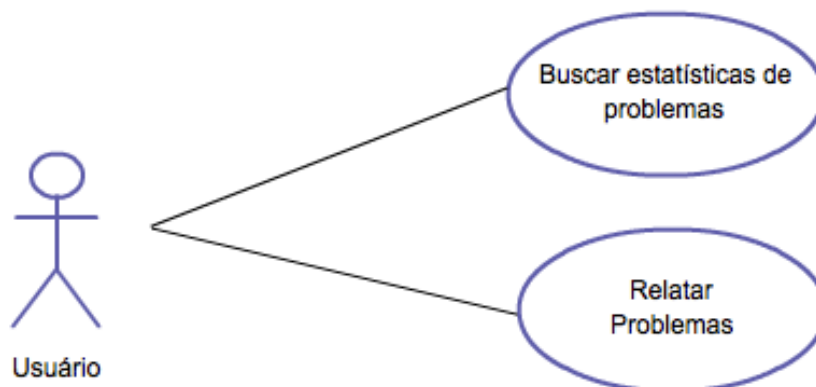


Figura 1. Diagrama de Casos de Uso.

O quadro seguinte apresenta o detalhamento do caso de uso *Buscar estatísticas de problemas*.

Caso de Uso – Buscar Estatísticas de Problemas		
Atores	Usuario	
Pré-Condições	Nenhuma	
Pós-Condições	Estatísticas de problemas que correspondam a localização geográfica do usuário.	
Descrição	Este caso de uso permite ao usuário obter remotamente as estatísticas de problemas identificados próximos a localização geográfica de seu dispositivo, em um raio de 03 quilômetros.	
Sequência de Eventos		
Usuario	Aplicativo	Servidor
1: O caso de uso inicia com o usuário acessando o aplicativo mobile.		
	2: Solicita permissão de acesso a localização geográfica do dispositivo.	
3: Usuário dá permissão ao aplicativo para ter acesso a sua localização geográfica.		
	4: Aplicativo obtém a localização geográfica do usuário e busca estatísticas de problemas de acordo com a localização geográfica do usuário.	
		5: Servidor retorna ao aplicativo as estatísticas de problemas com base na localização fornecida.
	6: Aplicativo exibe as estatísticas de problemas com base na localização do usuário.	

Na sequência é apresentado o detalhamento do caso de uso *Relatar Problemas*.

<b>Caso de Uso – Relatar Problemas</b>		
Atores	Usuario	
Pré-Condições	Nenhuma	
Pós-Condições	Publicar o relato de problema do usuário no serviço web PATROOL.com.	
Descrição	Este caso de uso permite ao usuário publicar o relato de um problema, utilizando dados da localização física do dispositivo móvel, no serviço web PATROOL.com.	
<b>Sequência de Eventos</b>		
<b>Usuario</b>	<b>Aplicativo</b>	<b>Servidor</b>
1: O caso de uso inicia com o usuário escolhendo a opção Relatar Problema no menu.		
	2: Solicita permissão de acesso a localização geográfica do dispositivo.	
3: Usuário dá permissão ao aplicativo para ter acesso a sua localização geográfica.		
	4: Aplicativo obtém a localização geográfica do usuário e exibe no mapa.	
5: Usuário informa os dados do problema (horário, tipo de problema e descrição do problema)		
	6: Aplicativo utiliza o protocolo HTTP para postar os dados no serviço PATROOL.com.	
		7: Servidor retorna a confirmação da postagem do problema no serviço remoto.
	8: Aplicativo exibe uma mensagem de sucesso para o usuário.	

### 3.2.2 Diagrama de Sequência

A figura abaixo apresenta a sequência de eventos do caso de uso *Buscar estatísticas de Problemas*.

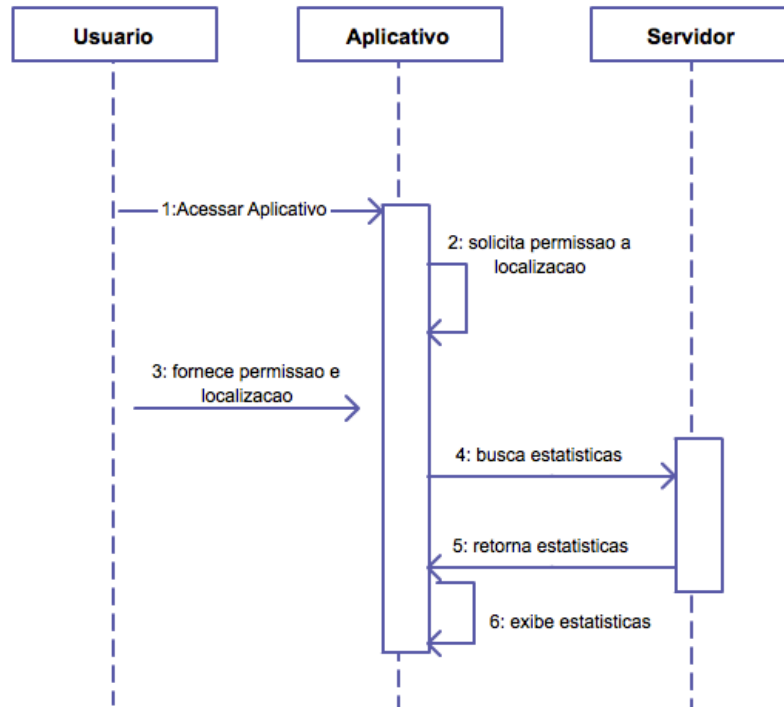


Figura 2. Diagrama de Sequência - Buscar Estatísticas de Problemas

A figura x3 demonstra o diagrama de sequência do caso de uso *Relatar Problemas*.

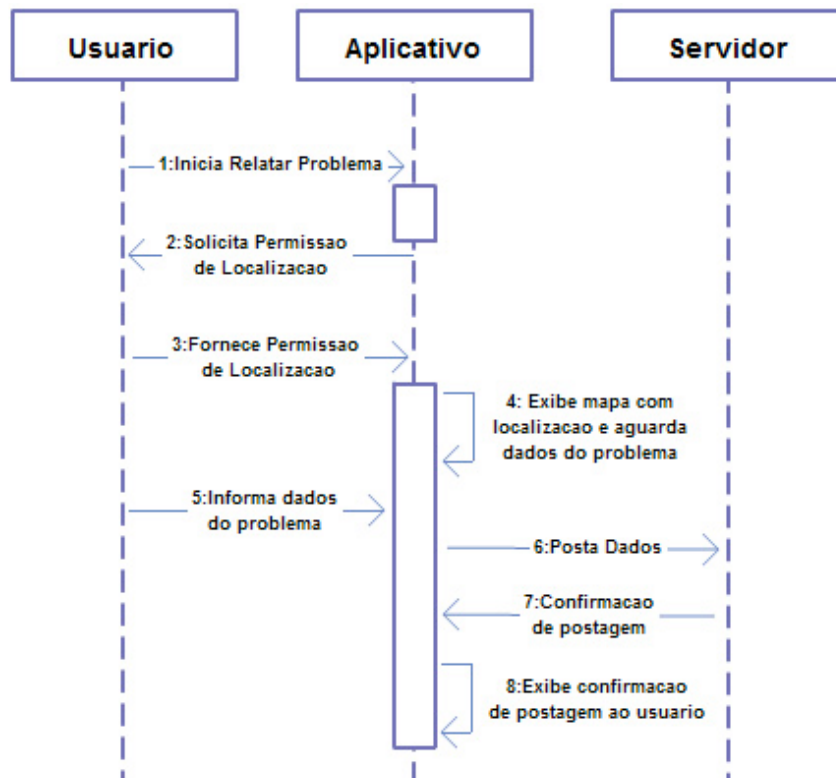


Figura 3. Diagrama de Sequência - Relatar Problemas

### 3.3 Implementação

Para a implementação do aplicativo foi utilizada a IDE *Titanium Studio* que é um ambiente para criar, gerenciar, desenvolver e fazer o debug de aplicações Titanium.

#### 3.3.1 Instalando o Titanium Studio

Para fazer o download do Titanium Studio é necessário fazer um cadastro no site Appcelerator (<http://www.appcelerator.com>), em seguida deve-se acessar o área do cliente em <http://my.appcelerator.com> para efetuar o download da IDE.

#### 3.3.2 Configurando o Titanium Studio

Logo após a aplicação Titanium Studio estiver sido instalada é necessário configurá-la. Para fazer isso, com o aplicativo aberto clique no logotipo da Appcelerator e em seguida clique em “*Configure*” como demonstra a Figura 3.

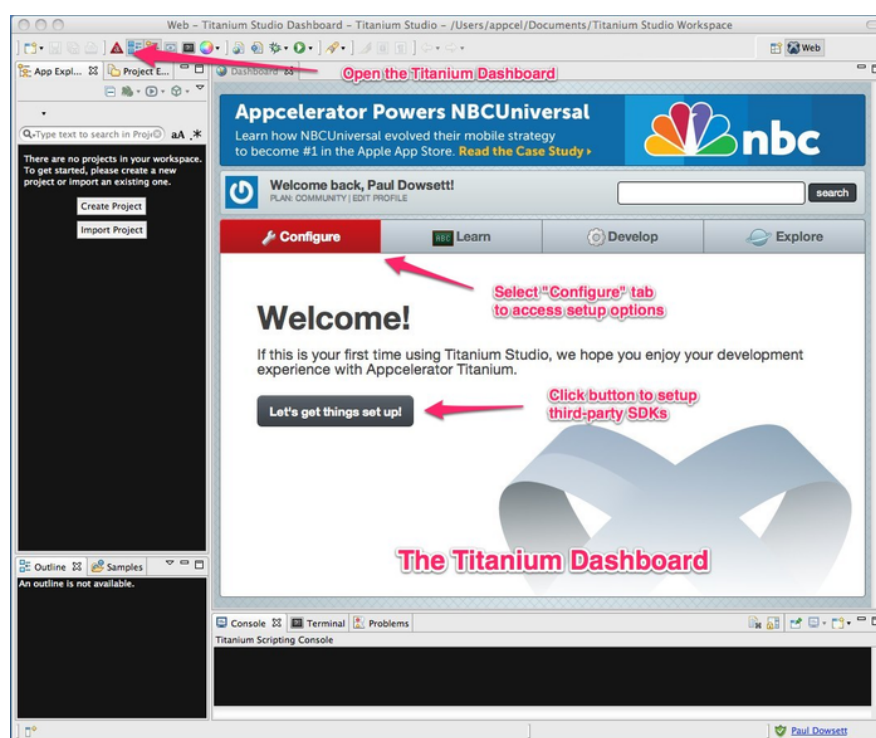


Figura 4. Configurando a IDE Titanium Studio (Fonte: <http://docs.appcelerator.com>)

Um aplicativo Titanium pode utilizar um vasto número de funcionalidades nativas da plataforma mobile, para tanto é necessário instalar as SDK's de cada plataforma. Nesse caso vamos instalar a SDK nativa para Android clicando no ícone

do Android e seguindo as instruções que aparecerão. Utilizaremos nesse trabalho a SDK do Android 2.2 (API 8).

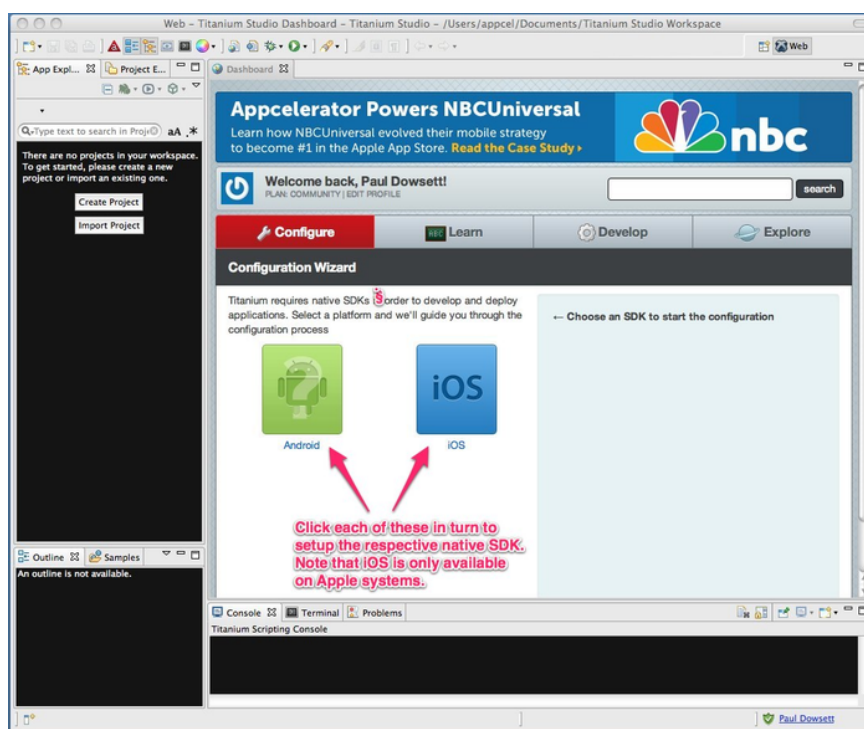


Figura 5. Instalando as SDKs através da IDE Titanium Studio (Fonte: <http://docs.appcelerator.com>)

## 3.4 Desenvolvendo o Aplicativo

### 3.4.1 Utilizando a Linguagem JavaScript

Os desenvolvedores web já possuem certa experiência com a programação JavaScript, mas a maioria está acostumada a utilizá-la para adicionar interatividade aos aplicativos, poucos tem utilizado para construir aplicações inteiras com JavaScript.

JavaScript é uma linguagem bastante flexível na medida em que é fracamente tipada. Isso significa que é possível reaproveitar objetos, variáveis e funções a qualquer momento no código através do método de atribuição. Mas esta flexibilidade deve ser tratada com responsabilidade. Se não tiver cuidado e seguir alguns princípios o poder expressivo de um código JavaScript pode causar tantos problemas quanto ele pode resolver.

Ao desenvolver um aplicativo utilizando JavaScript deve se esforçar para alcançar os seguintes princípios:

- Proteger o escopo global;
- Manter o código limpo;
- Utilizar um único contexto de execução.

### 3.4.2 Obtendo dados Remotamente

O aplicativo interage com o servidor remoto utilizando o protocolo HTTP através do objeto `Titanium.Network.HTTPClient`. A Figura 7 apresenta como as estatísticas de problemas são obtidas remotamente. O primeiro passo é instanciar o objeto `HTTPClient`. Deve-se criar uma função callback `onload`, que será responsável por processar os dados retornados, e também deve-se criar uma função callback `onerror`, que será responsável pela lógica de tratamento de erros.

```
(function() {
    pt.service = {};

    //Carrega dados
    pt.service.loadData = function(lat, lng) {

        //HTTPClient
        var xhr = Titanium.Network.createHTTPClient();

        //Sucesso - Processa o XML
        xhr.onload = function() {
            //Logica de Processamento do XML
        };

        //Em caso de Erro
        xhr.onerror = function() {
            //Logica de tratamento de Erro
        };

        //Abre conexao para recuperar dados
        xhr.open('GET',
            'http://patrool.com/service/get-estatisticas-problema-local-xml/lat/'+
            lat+'/lng/'+lng);

        //Finalmente faz requisicao de dados remotos
        xhr.send();

    };
})();
```

Figura 6. Trecho do código-fonte mostrando como se obtém dados remotos.

### 3.4.3 Obtendo estatísticas da localização

A SQL responsável por buscar as estatísticas de problemas de determinada local é apresentada na Figura 8. Deve-se notar que no comando SQL há uma fórmula matemática, que é alimentada com a latitude e longitude da localização do usuário, para calcular a distância em metros do usuário dos problemas cadastrados.

Nesse caso, só são contabilizados os problemas que estão a uma distância máxima de 3 (três) mil metros da localização do usuário.

```

SELECT
  c.categoria
  ,COUNT(c.categoria) AS qtd
FROM
  (SELECT
    (CASE
      WHEN (categoria = 1 || categoria = 2 || categoria = 3 ||
            categoria = 4 || categoria = 5) THEN categoria
      ELSE 0
    END) AS categoria
    ,( 6371 * acos( cos( radians({$lat}) ) * cos( radians( lat ) ) *
      cos( radians( lng ) - radians({$lng}) ) + sin( radians({$lat}) )
      * sin( radians( lat ) ) ) ) ) * 1000 AS distancia
    FROM
      ocorrencias o
    HAVING
      distancia < 3000
  ) AS c
GROUP BY
  (c.categoria)

```

Figura 7. SQL que contabiliza os problemas próximos a localidade do usuário.

### 3.5 Publicando um problema no serviço PATROOL

O aplicativo interage com o servidor remoto utilizando o protocolo HTTP através do objeto `Titanium.Network.HTTPClient`. A Figura 7 apresenta como as estatísticas de problemas são obtidas remotamente. O primeiro passo é instanciar o objeto `HTTPClient`. Deve-se criar uma função callback `onload`, que será responsável por processar os dados retornados, e também deve-se criar uma função callback `onerror`, que será responsável pela lógica de tratamento de erros.

### 3.6 Apresentação do Aplicativo Mobile

A seguir, na Figura 9, é apresentado como o aplicativo desenvolvido se comporta sendo executado em um smartphone Android (versão 2.2).





Figura 8. Aplicativo exibido a localização do usuário e as estatísticas de problemas da região.



Figura 9. Aplicativo exibindo o relato de um problema.



Figura 10. Inserção da descrição do problema.

## 4 CONCLUSÃO

Appcelerator Titanium ajuda desenvolvedores a aproveitar suas habilidades em JavaScript para construir aplicativos nativos móveis rapidamente que rodam em múltiplas plataformas. Ele fornece ferramentas para construir aplicativos que se comportam como nativos, se encaixando perfeitamente dentro do ecossistema de cada plataforma.

Titanium permite criar aplicativos para Android que se sentem como aplicativos do Android. É possível criar menus nativos, colocar mensagens de status na barra de navegação Android e acessar o hardware do dispositivo. É possível ainda iniciar *Activities* e lançar ou receber *Intents*. Com o aplicativo finalizado é possível também publicar o aplicativo no *Google Play*, antigo *Android Market*.

### 4.1 Trabalhos Futuros

Pretende-se aprimorar o aplicativo mobile com diversas outras funcionalidades do serviço PATROOL.com, como por exemplo, a possibilidade de capturar imagens no momento de relatar um problema.

Em decorrência da versatilidade da plataforma Titanium pretende-se também disponibilizar o aplicativo para a plataforma iOS. E, naturalmente, publicar o aplicativo na *App Store* do *iTunes*, já que o Titanium cumpre as diretrizes da *Apple* na construção do aplicativo.

Na sequência são apresentadas quatro figuras que ilustram como o aplicativo deverá se comportar futuramente.



Figura 11. Visualização de índices de relatos de determinada região.



Figura 12. Visualização de relatos em lista



Figura 13. Visualização de relatos em mapa



Figura 14. Visualização de detalhes de relato

## 5 REFERÊNCIAS

**About Google Maps.** Disponível em:

<http://support.google.com/maps/bin/answer.py?hl=en&answer=7060>

Acesso em: 18/09/2012, 18h38min.

**Appcelerator | Titanium Mobile Development Platform.** Disponível em:

<http://www.appcelerator.com/>

Acesso em: 18/00/2012, 18h41min.

**BNAPP ebook – Appcelerator Titanium Mobile.** Disponível em:

[http://docs.appcelerator.com/titanium/2.1/index.html#!/guide/BNAPP\\_ebook](http://docs.appcelerator.com/titanium/2.1/index.html#!/guide/BNAPP_ebook)

Acesso em: 18/09/2012, 18h42min.

**Creately.** Disponível em:

<https://creatly.com>

Acesso em: 18/09/2012, 18h40min.

**FAQ Google Maps API – Google Developers.** Disponível em:

[https://developers.google.com/maps/faq?hl=pt-BR#tos\\_commercial](https://developers.google.com/maps/faq?hl=pt-BR#tos_commercial)

Acesso em: 18/09/2012, 18h35min.

**Geolocation API Specification.** Disponível em:

<http://www.w3.org/TR/geolocation-API/>

Acesso em: 18/09/2012, 18h36min.

**Google Maps – Wikipedia.** Disponível em:

[http://en.wikipedia.org/wiki/Google\\_Maps](http://en.wikipedia.org/wiki/Google_Maps)

Acesso em: 18/09/2012, 18h39min.

KING Chris; SEN Robi; ORTIZ C. Enrique; 2011: **Android in Action.** Manning Pubns Co 3ª Edição.

PENDER, Tom; 2004, **UML A Bíblia**. Campus 2ª Edição.

POLLENTINE, Boydlee, 2011: **Appcelerator Titanium Smartphone App Development Cookbook**. Packt Publishing 1ª Edição.