

# Sistema de Gerenciamento de manutenção de pontos de iluminação pública CidadeIluminada

Arthur de Paula Bressan  
(IBTA)

Thais Matias Nogueira  
(IBTA)

2015

## Resumo

Este trabalho apresenta um estudo sobre dispositivos móveis, explicando a implementação de um protótipo de software que auxiliará no recebimento e análise de denúncias relacionados a iluminação pública na cidade de São José dos Campos. O aplicativo foi desenvolvido em Android e será responsável por enviar estas denúncias para um ambiente web. A aplicação web será responsável por responder às requisições do aplicativo Android e processá-las de modo que a manutenção dos pontos de iluminação pública seja feita de uma maneira mais rápida e menos burocrática.

**Palavras-chave:** Iluminação pública. Aplicativo Android. Sistema web para manutenção.

## Introdução

A Iluminação pública é um instrumento de cidadania, pois é um item essencial à qualidade de vida e a segurança dos centros urbanos, ajudando a preservar e embelezar os patrimônios públicos e a prática de diversas atividades noturnas no espaço público.

Segundo (AVER, 2015) a falta de iluminação contribui consideravelmente com o aumento dos índices de criminalidade e a falta de segurança dos cidadãos que por algum motivo tem que transitar no período noturno por pontos sem iluminação, ou com iluminação precária. Estudos feitos pela organização Campbell Collaboration (NEWTON, 2015), mostraram que a melhoria na iluminação pública em algumas cidades do Reino Unido e EUA, ajudaram a reduzir a criminalidade em até 21%, e concluiu que esta melhoria promoveu o aumento da vigilância da comunidade, que atuou como elemento inibidor contra a criminalidade local.

Os responsáveis pela instalação e manutenção da iluminação pública dos municípios no Brasil mudaram em dezembro de 2014. Até então, elas eram feitas pelo estado, normalmente através das concessionárias de energia elétrica. Agora é a própria prefeitura que tem a responsabilidade por garantir as lâmpadas acessas. Essa mudança foi ocasionada pela Resolução Normativa 414 (ANEEL, 2015), uma determinação constitucional expedida

pela ANEEL (Agência Nacional de Energia Elétrica) que obrigou todos os 5.570 municípios brasileiros a se responsabilizar pela manutenção da infraestrutura da iluminação pública.

A cidade de São José dos Campos/SP conta hoje com cerca de 75.000 pontos de iluminação distribuídos pela cidade. A média de lâmpadas com defeito é cerca de 900 por mês, de acordo com a Secretaria de Obras da cidade (OBRAS, 2015). Um dos maiores problemas a serem enfrentados é o recebimento de chamados para solucionar problemas com os postes defeituosos, já que o canal de reclamações da prefeitura engloba todos os problemas da cidade, causando assim, uma sobrecarga no atendimento.

A Urbanizadora Municipal (URBAM) foi a empresa contratada pela prefeitura de São José dos Campos para realizar a manutenção da iluminação pública do município. Com a proposta de auxiliar a URBAM no encontro dos pontos com problemas e agilizar as respostas aos chamados abertos pelos munícipes, o trabalho “Cidade Iluminada” apresenta uma solução de um aplicativo para smartphones e uma aplicação Web para a gestão das solicitações de manutenção dos pontos de iluminação pública.

O atual processo de recebimento de chamados, funciona de maneira manual, onde um funcionário deve entrar em um sistema do canal de recebimentos de denúncias, o 156, e coletar manualmente todas as denúncias recebidas no dia, colocar em uma planilha no Excel, e separar por regiões. Após este passo, a planilha já dividida em regiões, são checados os chamados um a um caso haja duplicidade. Uma vez que todos os pedidos já estão na planilha e caso não haja duplicidade, uma ordem de serviço é feita e enviada a URBAM para que a manutenção seja feita. Quando a manutenção é feita, esta ordem de serviço retorna e são retirados da planilha os chamados que foram atendidos.

Evidentemente é um processo muito complexo, e passível a erros, onde por um menor descuido vários detalhes poderão ser perdidos, como denúncias duplicadas que irão gerar o retrabalho quando a ordem de serviço é gerada. Por isso nós propomos uma solução, onde não será mais necessária a coleta e verificação manual de todas as denúncias. Todo este trabalho excessivo será reduzido e em poucos cliques toda a análise será feita automaticamente.

Nossa missão é disponibilizar uma ferramenta que irá aperfeiçoar o recebimento dos chamados relacionados aos postes de iluminação e viabilizar a comunicação do cidadão diretamente com a entidade competente através de uma aplicação Android e um ambiente Web onde essas informações serão processadas. Neste aplicativo, o usuário poderá registrar o problema na rede pública de iluminação, enviar uma foto e a localização do problema e acompanhar o andamento do protocolo aberto por ele. No ambiente Web, este protocolo será analisado e separado de acordo com a região, bairro, rua e tipo de problemas, também serão analisados se não há duplicidade em algum ponto. E além disso, o sistema tem como função preparar a ordem de serviço para a manutenção, e fazer o controle dos materiais utilizados.

## 1 Fundamentação Teórica

### 1.1 Engenharia de Software

Segundo Pressman, a engenharia de software é definida como: “O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.”(PRESSMAN, 2011)

O termo engenharia de software foi criado por volta da década de 60, com o objetivo de sanar a crise do software e contornar os problemas oriundos do grande crescimento da demanda de softwares e da complexidade a eles imposta, assim padronizando o desenvolvimento deixando-o mais controlado e sistemático.

Segundo Breitman “o desenvolvimento de sistemas de software é uma atividade complexa que envolve um grande número de recursos, coordenados de modo a atingir um mesmo objetivo”. (BREITMAN, 2000) Por isso esta tarefa exige um planejamento detalhado e documentado por se tratar de um objeto dinâmico que está sempre se adaptando as necessidades dos usuários de acordo com as ferramentas usadas no seu desenvolvimento

Para Pressman, a engenharia de software tem três fases: definição, desenvolvimento e manutenção (PRESSMAN, 2011).

A fase de definição, estabelece as propriedades, características, limitações e funcionalidades do software. Esta fase é dividida em três etapas, a análise do sistema onde são definidas as funções que o sistema deve apresentar. O planejamento de projeto, onde são analisados os riscos, custos e recursos que serão necessários no desenvolvimento. E por fim a análise dos requisitos, nesta etapa são verificados os requisitos necessários para que atenda as funções do sistema.

A fase de desenvolvimento, é a parte onde os desenvolvedores transformam as especificações em código. O projeto é analisado, estabelecendo a arquitetura e estrutura dos dados, seguida pela codificação numa linguagem de programação que atenda as necessidades do projeto.

E finalmente a fase de manutenção dedica-se a aprimorar ainda mais o sistema, sendo validado pelo usuário final, que nem sempre está satisfeito. Algumas mudanças podem ocorrer no projeto final, como correções pontuais que acontecem quando usuário acha um erro, adaptações caso novos atributos precisarem ser adicionados, e por fim o aprimoramento funcional, que acontece a medida que o usuário se acostuma a utilizar o sistema, e poderá sugerir novas funcionalidades.

## 1.2 Método de desenvolvimento Agile, Kanban e Scrum

O Agile foi publicado em 2001 por meio da publicação de um manifesto por Jeff Sutherland, Ken Schwaber et. al. que priorizava a rápida adaptação às novas situações, a colaboração entre si dos times de desenvolvimento, a colaboração dos times de desenvolvimento com os seus clientes e a entrega de softwares que estejam funcionando. (SUTHERLAND, 2015)

Apesar do Agile não ditar como o desenvolvimento deva ser feito por si só, ele é um conjunto de conceitos para que métodos, como o Kanban e o Scrum, sejam construídos.

O Scrum consiste basicamente em separar atividades de trabalho em sprints, com o objetivo de entregar todas as atividades da forma mais completa possível, trabalhando assim de modo incremental a cada sprint até que o cliente decida que o software esteja com uma qualidade suficiente para ser entregue. Além disso, seguindo os princípios dos métodos Agile, o Scrum ajuda na colaboração intra-equipe por meio de reuniões stand-up diárias, uma reunião pós-sprint de retrospectiva e uma reunião pré-sprint para planejamento do próximo sprint. Para facilitar o planejamento, existem somente três “papéis” no Scrum: O Dono do Produto (Product Owner), o Mestre do Scrum (Scrum Master) e a Equipe (Team).

A Equipe consiste nos desenvolvedores que irão efetivamente fazer a implementação do produto baseado nas necessidades do Dono, que é responsável por demonstrar os seus requisitos. O Mestre do Scrum é responsável por além de transformar os requisitos do Dono em atividades que podem ser divididas entre os membros da Equipe, é responsável principalmente por gerenciar a equipe para que as atividades sejam cumpridas em cada sprint. ([ALLIANCE, 2015](#))

O Kanban é uma técnica que separa as atividades em estados pré-definidos pela equipe que a utiliza, e é utilizada principalmente para visualizar o progresso de diferentes atividades dentro de um contexto. Por si só, o Kanban não diz como introduzir novos processos, mas sim como montar os processos existentes, como o Scrum, de forma visual.

### 1.3 Resolução Normativa 414

A Agência Nacional de Energia Elétrica (Aneel), determinou que a partir de janeiro de 2015, a responsabilidade da manutenção da iluminação pública será do município. Esta resolução vem de uma determinação da Constituição Federal de 1988 que, em seu artigo 149-A e conforme Emenda Constitucional número 39, de 19 de dezembro de 2002. A partir desta data limite, o município terá que arcar também com os investimentos para a ampliação e modernização da rede, instalação de medidores de acordo com a concessionária e também a manutenção do cadastro de ativos georreferenciados.

A responsabilidade da manutenção da rede até então era das concessionárias de distribuição de energia elétrica locais. E segundo um sondagem da Confederação Nacional dos Municípios (CNM), esse transferência poderá aumentar aproximadamente até 28% a despesa anual dos municípios. ([LIMA, 2015](#))

### 1.4 Desenvolvimento mobile

Durante as últimas décadas foi notável o aumento significativo do uso de dispositivos móveis, segundo o grupo Ibope ([AMADOR, 2015](#)), com diversas funções e finalidades específicas, como computadores de bolso, smartphones, telefones celular, consoles portáteis combinado com as mais diversas tecnologias como GPS, navegadores de Internet, WAP, leitores de áudio, vídeo e texto, entre outros, e juntamente com a evolução das tecnologias e o aumento do poder computacional tornou-se possível o desenvolvimento de aplicações independentes do hardware ou software do dispositivo.

Na América Latina não foi diferente, até o ano de 2014 o Brasil possuía cerca de 137 milhões de usuário, um pouco mais de 67% da população, segundo pesquisa conduzida pela eMarketer ([EMARKETER, 2015](#)), que ainda prevê um aumento de mais 4 milhões de usuários até o final de 2015.

Com base nesses dados é irrefutável o crescimento do mercado neste segmento o que propicia muitas oportunidades no desenvolvimento de aplicativos mobile. É visível a necessidade de encontrar novas soluções para os usuários de smartphones, e integrar novas tecnologias a este mercado, pois a necessidade não para de crescer junto com número de usuários.

## 2 Tecnologias

As tecnologias utilizadas foram escolhidas de acordo com o seu nicho específico, tendo em mente que cada uma delas cumpre um papel importante para entregar uma

solução de qualidade para o cliente.

## 2.1 Python

Python é uma linguagem de programação multiparadigma (se destacando os paradigmas orientado a objetos e funcional) , interpretada, imperativa e de alto nível; possui tipagem dinâmica e código fonte livre (PSF, 2010) e gratuito, que pode ser facilmente encontrado na Internet e podendo ser livremente modificado e redistribuído, e também possui fácil integração com outras plataformas como .NET, Java e C/C++.

A linguagem foi originalmente concebida e implementada pelo holandês Guido Van Rossum no início da década de 1990 (PSF, 2010), e a ideia original era de criar um sucessor para a linguagem ABC.

Uma das grandes vantagens de utilizar o Python é devido à comunidade que surgiu ao redor da linguagem, ocasionando portanto, em um grande número de bibliotecas (cerca de 70,000 pacotes, de acordo com o repositório central Python Package Index)(PYPI, 2010) tanto open-source como bibliotecas de livre utilização.

O Python foi escolhido para a implementação, junto com o framework web Flask, devido a sua facilidade e velocidade de implementação, além de experiência anteriores com o Python e com o framework Flask.

### 2.1.1 Flask

Flask é um microframework para desenvolvimento web, criado em Python baseado em duas ferramentas Werkzeug, como servidor de aplicação e Jinja2, como o motor de renderização. O Flask mantém o núcleo simples e, para isso, ele suporta extensões que são capazes de adicionar diversas funcionalidades a aplicação como por exemplo suporte a banco de dados, camadas de cache, geração de formulários, etc.. (FLASK, 2010)

Existem vários outros frameworks de desenvolvimento web para Python, como por exemplo o Django, Pyramid, Tornado, entre outros. O Flask foi escolhido devido a sua grande capacidade de extensão e por causa de experiências anteriores com o desenvolvimento de aplicativos Web.

### 2.1.2 SQLAlchemy e Flask-SQLAlchemy

O SQLAlchemy é um conjunto de ferramentas para operar com SQL no Python de forma abstrata, englobando assim um ORM e um conjunto de ferramentas para a utilização do SQL no Python de forma natural para os usuários de Python. (SQLALCHEMY, 2010)

Além disso, o SQLAlchemy abstrai os principais bancos de dados, como o MySQL, PostgreSQL, Oracle e até mesmo SQLite. Isso permite a codificação do acesso ao banco de dados da aplicação sem precisar depender de qual banco específico.

A integração do Flask com o SQLAlchemy se dá por meio da extensão Flask-SQLAlchemy, e ela tem como objetivo principal facilitar a configuração do SQLAlchemy para a sua utilização em aplicativos Flask. (FLASK-SQLALCHEMY, 2010)

O SQLAlchemy e o Flask-SQLAlchemy foram escolhidos devido a sua grande flexibilidade e poder expressivo dentro de sua linguagem, além de sua fácil integração com o Flask.

### 2.1.3 Flask-Migrate e alembic

O alembic é uma ferramenta que trabalha junto com o SQLAlchemy para a geração e execução de migrações de banco de dados. Isso permite que alterações feitas no código das classes sejam refletidas na estrutura de tabelas no banco de dados de forma semi-automática. (ALEMBIC, 2010)

O Flask-Migrate é a biblioteca de integração do alembic com o Flask. Isso expõe ao Flask comandos que facilitam o gerenciamento destas migrações. (FLASK-MIGRATE, 2010)

O alembic foi escolhido não somente pela sua fácil integração com tanto o SQLAlchemy e o Flask, mas também por ele ser uma ferramenta de gerenciamento de migrações de alta qualidade.

### 2.1.4 Flask-Admin

O Flask-Admin é uma extensão do Flask que auxilia a criação de páginas de gerenciamento do back-end de aplicativos em Flask. Isso é feito expondo aos programadores utilidades que geram telas administrativas genéricas para que então cada aplicação as customize de acordo com suas necessidades. (FLASK-ADMIN, 2010)

O Flask-Admin foi escolhido para ser a base das telas do sistema de gerenciamento de ordens de serviço pois grande parte das funcionalidades básicas necessárias para um sistema CRUD já estão maduras no Flask-Admin.

### 2.1.5 Flask-Script

O Flask-Script disponibiliza um conjunto de ferramentas e classes em Python para o gerenciamento da aplicação Flask via linha de comando.

Inicialmente, o Flask-Script disponibiliza comandos para o gerenciamento do servidor de desenvolvimento e disponibiliza estruturas para que cada aplicação possa implementar seus próprios comandos. (FLASK-SCRIPT, 2010) Por exemplo a extensão Flask-Migrate disponibiliza comandos via o Flask-Script para preparar e executar migrações de banco de dados.

O Flask-Script foi escolhido para realizar a integração pois ela atende satisfatoriamente à demanda de auxílio no gerenciamento da aplicação.

### 2.1.6 xhtml2pdf

O xhtml2pdf é uma biblioteca que tem como objetivo transformar páginas HTML preparadas para o formato PDF. (XHTML2PDF, 2010)

A biblioteca foi utilizada para gerar as tabelas das ordens de serviço para comparar com a URBAM os endereços dos postes de cada ordem de serviço gerada.

Essa biblioteca foi utilizada pois ela era a mais conveniente de ser utilizada dado o fato que o Flask contém um renderizador de templates HTML.

### 2.1.7 fabric e fabtools

O fabric é uma utilidade que tem como objetivo executar comandos de gerenciamento de servidores via SSH. A grande vantagem de sua utilização é automatizar tarefas de

lançamento da aplicação ou até mesmo a preparação de ambientes de produção.([FABRIC, 2010](#))

O fabtools é uma biblioteca que expande o fabric para que a escrita de scripts fabric seja ainda mais fácil. Para isso, o fabtools expõe comandos para as ferramentas mais utilizadas, como comandos do sistema de arquivos, gerenciamento de pacotes de instalações, git, banco de dados, etc.. ([FABTOOLS, 2010](#))

O fabric e fabtools foram escolhidos para serem utilizados devido experiências positivas anteriores de gerenciamento de servidores.

#### 2.1.8 unicodecsv

O módulo csv do Python é disponibilizado na biblioteca padrão do Python para a leitura e escrita de arquivos separados por vírgula. Porém, devido às dificuldades do Python 2.7 para a leitura e uso de textos com codificação em Unicode, surgiram várias bibliotecas que vieram para suplantiar os módulos da biblioteca padrão.

Uma dessas bibliotecas é justamente o unicodecsv, que simplesmente faz o tratamento de textos em Unicode para que não ocorra erros na leitura e escrita desses arquivos com a codificação Unicode. ([UNICODECSV, 2010](#))

Ela foi escolhida pois, além de atender perfeitamente essa demanda, ela tem uma grande facilidade no seu uso, pois a sua interface é idêntica ao módulo csv padrão do Python

#### 2.1.9 Flask-Security

O Flask-Security é a integração de uma série de bibliotecas do Flask relacionadas ao controle de acesso de usuários com o objetivo de facilitar a implementação desse tipo de sistema. ([FLASK-SECURITY, 2010](#))

Apesar de sua baixa flexibilidade, o Flask-Security foi escolhido para ser usado no controle de acesso pois ele atendia plenamente as necessidades do projeto na questão de gerenciamento de usuários e controle de acesso.

### 2.2 Java

A linguagem de programação Java foi lançada em 1995, e, desde o início de seu desenvolvimento em 1993 ela foi especificada para ser “independente de plataformas, porém segura o suficiente para atravessar redes e poderosa o suficiente para substituir código nativo”. ([NIEMEYER, 2013](#))

O Java teve as mais diversas aplicações, sendo usado para applets para a Web, para programas desktop, para serviços em servidores, para aplicações Web e até rodando em sistemas embarcados. Essa última aplicação provavelmente possibilitando a implementação do sistema operacional para smartphones Android.

O Java foi utilizado para a implantação do aplicativo Android do Cidade Iluminada pois é a linguagem que o sistema operacional Android usa nativamente para as aplicações, tornando assim o desenvolvimento do aplicativos de acordo com os padrões de desenvolvimento Android.



### 2.2.1 Android

O sistema operacional Android foi lançado pela Google no ano de 2007, e no ano de 2008 foi lançado uma API para os desenvolvedores. Desde o seu lançamento, este sistema operacional obteve grande popularidade em seu uso, chegando a ter mais de 89% do mercado brasileiro no primeiro trimestre de 2015, segundo o site Kantar Worldpanel ComTech. (COMTECH, 2010)

A Google provê uma API de fácil manipulação, com código fonte aberto baseada em Java para incentivar e simplificar o desenvolvimento de aplicações, o que tem sido uma boa estratégia já que o número de aplicativos disponíveis no Android Market chegou a 1.6 milhões no último semestre de 2015.

O sucesso do Android no mercado brasileiro, que segundo a pesquisa do site The Statistics Portal, o market share do sistema operacional chegou a mais de 77% no primeiro semestre de 2015 (STATISTA, 2010), e a facilidade de desenvolvimento foram os principais motivos da escolha do uso deste para o desenvolvimento do aplicativo Cidade Iluminada.

### 2.2.2 GSON

GSON é uma biblioteca do Google para a serialização e deserialização de objetos Java arbitrários. O objetivo desta biblioteca, além de prover essas conversões, é trabalhar com objetos que o programador não necessariamente tem o código-fonte disponível, mantendo uma interface simples (métodos fromJson e toJson) para quando for precisar converter de e para JSON. (GSON, 2015)

A biblioteca foi utilizada pois ela atende a necessidade principal de tratar JSON mas ainda sendo relativamente fácil de utilizar e configurar as classes que precisam dessa conversão.

### 2.2.3 Joda-Time

A biblioteca Joda-Time tem como objetivo prover facilidades para o programador Java manipular objetos de data e tempo, tendo em vista que a biblioteca padrão do Java deixa a desejar no aspecto de facilidade de uso.

Ela foi utilizada para a geração de timestamps em UTC para marcar o envio das requisições do aplicativo e para gerar os nomes de arquivos das fotos tiradas pelo aplicativo. A biblioteca Joda-Time foi escolhida para substituir a biblioteca padrão do Java nas suas próximas versões. (JODA-TIME, 2015)

### 2.2.4 Retrofit

A biblioteca Retrofit tem como objetivo disponibilizar ferramentas para que se possa definir APIs REST como interfaces Java para serem então usadas no código da aplicação sem maiores dificuldades. Além disso, ela disponibiliza modos de fazer requisições síncronas ou assíncronas, dependendo das necessidades de cada aplicação. (RETROFIT, 2015)

A biblioteca Retrofit foi utilizada devido a sua facilidade de utilização e adaptação à API já definida do trabalho. Houve uma primeira tentativa de implementação utilizando a biblioteca Unirest, mas devido a dificuldades de compilação da biblioteca para utilização no ecossistema Android a Unirest foi substituída pela Retrofit.



### 2.2.5 Sugar

Sugar é uma biblioteca que tem como objetivo de ser uma ORM especificamente para aplicações Android, tendo uma interface por objetos e classes com o banco de dados SQLite do dispositivo. Além disso, ela tem a pretensão de ser simples de usar e simples de configurar. Além disso, ela possui suporte à migrações de banco automáticas. (SUGAR, 2015)

O Sugar foi escolhido justamente devido a sua simplicidade tanto na configuração quanto no seu uso dentro do código.

## 2.3 JavaScript

JavaScript é uma linguagem dinâmica, funcional e orientada a objetos, suas aplicações são distribuídas somente em forma de código fonte, e é executada em todos os navegadores modernos com poucas ou nenhuma diferença de plataforma. Isso é especialmente importante nos dias atuais pois até mesmo smartphones têm o seu próprio navegador, abrindo então uma nova plataforma inteira de aplicações

Em 1994, foi fundada uma empresa chamada Netscape, e um dos principais objetivos dela foi a exploração da Web nascente, e sua então nova necessidade de serem disponibilizadas páginas dinâmicas. Um exemplo disso foi a necessidade de validação de formulários: O usuário teria que preencher todo o formulário, enviá-lo e então receber o retorno do servidor se falhou ou não. Em 1995, em uma parceria com a Sun, a Netscape decidiu que esta nova linguagem teria que ter uma sintaxe parecida com a linguagem Java, descartando assim a linguagem Python (entre outras) como candidatas para ser implementadas em seu navegador.

Brendan Eich foi contratado para implementar a linguagem Scheme no navegador, porém, por causa dessa decisão da Netscape, ele acabou implementando a linguagem que seria conhecida como JavaScript (o nome oficial é ECMAScript) em 10 dias. (RAUSCHMAYER, 2014)

Apesar de seu surgimento rápido, a linguagem foi passando por várias alterações que a deixam muito mais poderosa e mais em linha com as necessidades atuais de desenvolvimento para a Web moderna.

O JavaScript foi escolhido para a implementação de páginas Web dinâmicas no projeto principalmente pelo fator de ele ser suportado pelos principais navegadores existentes no mercado, como o Mozilla Firefox, o Google Chrome e o Internet Explorer, sendo os dois primeiros softwares livres para a utilização em ambientes pessoais ou corporativos.

### 2.3.1 JSON

JSON (JavaScript Object Notation) é um formato para guardar e transmitir dados em formato de texto. O JSON foi descoberto e formalizado por Douglas Crockford em 2001. (CROCKFORD, 2006) Desde então vem ganhando popularidade como um formato de troca de dados entre aplicações web. Existem bibliotecas para a leitura do formato JSON para a maioria das linguagens de programação, além do suporte praticamente nativo no JavaScript e Python.

O JSON foi escolhido como método de troca de dados pois ele tem suporte praticamente nativo em JavaScript e Python, além de ser mais eficiente em termos de quantidade de bytes enviados do que XML para envio de dados mais simples.

### 2.3.2 AngularJS

AngularJS é um framework do tipo MVC (Model-View-Controller) JavaScript que tem como objetivo facilitar o desenvolvimento web, estendendo e adaptando o HTML para melhorar a experiência de desenvolvimento de páginas Web altamente dinâmicas. Esta framework foi criada em 2009 por Miško Hevery e Adam Abrons, e é open-source. ([ANGULARJS, 2006](#))

O framework foi utilizado na criação da página de controle dos protocolos vindos da aplicação Android, sendo parte fundamental da implementação do módulo de tempo real desta página.

O AngularJS foi escolhido devido a sua grande capacidade de facilitar o desenvolvimento e manutenção de páginas Web altamente interativos.

### 2.3.3 JQuery

JQuery é uma biblioteca JavaScript que tem como objetivo principal facilitar a seleção e manipulação de documentos HTML e o controle de eventos nas páginas. Isso possibilita a criação de páginas interativas em JavaScript com muito menos dificuldades. Além disso, o JQuery disponibiliza uma API única para todos os navegadores principais. ([JQUERY, 2006](#))

O JQuery foi utilizado em várias das páginas da parte de gerenciamento de protocolos e serviços, pois ela se tornou uma peça fundamental para o desenvolvimento Web moderno. Das páginas utilizadas, destaca-se a página de criação de Ordens de Serviço, onde a parte de seleção de postes foi implementada em JQuery.

## 2.4 Representational State Transfer (REST)

O REST é um estilo de arquitetura para aplicações Web proposta por Roy Thomas Fielding em sua dissertação de PhD. ([FIELDING, 2000](#)) Esta tese tinha como objetivo de, além de propor esse tipo de arquitetura, avaliar se serviços disponíveis na Web até então seguiam os quesitos seguintes:

- **Cliente-Servidor:** A aplicação teria que manter independentemente suas implementação de clientes (interface gráfica, por exemplo) da implementação dos serviços oferecidos por ela (banco de dados, por exemplo)
- **Sem estado:** Isso significa que, a cada requisição enviada para o servidor, esta teria que incluir todos os dados para que a própria seja entendida, sem que o servidor precise depender de informações que por ventura ele possa ter armazenado anteriormente.
- **Cacheável:** Cada resposta a uma requisição tem que deixar explícito se é permitido guardar a resposta em um cache no cliente. A ideia é que, se o cliente precisar refazer uma requisição equivalente, esta requisição deveria utilizar o cache ao invés de ser enviada para o servidor.
- **Interface uniforme:** De acordo com Fielding em sua dissertação, é a característica fundamental que aplicações REST têm de implementar. Ela consiste em padronizar todos os componentes para que as implementações de clientes e serviços sejam simplificados.

- **Sistema em camadas:** Cada componente deve não conhecer ou depender de componentes de que este componente não precise interagir. A principal vantagem, de acordo com Fielding, é que o sistema em camadas pode ser usado para encapsular clientes e serviços legados
- **“Code-on-demand”:** A aplicação REST teria que oferecer um conjunto de recursos que viriam como código a ser executado no cliente. Este código seria como uma interface mínima da aplicação para o cliente consumir. A principal vantagem de implementar o Code-on-demand seria que, a parte básica da API não precisaria ser reimplementada pelo cliente. É o único quesito opcional para que uma interface fosse considerada como REST.

### 3 Desenvolvimento

O desenvolvimento do trabalho pode ser separado em duas fases ou períodos distintos, sendo estes o desenvolvimento do aplicativo para smartphones Android e o sistema de gerenciamento dos protocolos deste, e o desenvolvimento do sistema de gerenciamento de pontos de iluminação e de seus serviços e chamados.

#### 3.1 Aplicativo Android e seu sistema de gerenciamento

Para o desenvolvimento do aplicativo para smartphones Android e seu sistema de gerenciamento, foram aplicadas os métodos Agile de desenvolvimento Scrum e Kanban.

As atividades a serem cumpridas foram determinadas no início do projeto, e foram divididas em três categorias, e priorizadas dependendo da importância de cada atividade para o cumprimento do projeto.

Em relação à execução das atividades, elas foram divididas em 5 “sprints”, que foi um período de 14 dias em que as atividades selecionadas para fazer parte do sprint teriam que ser cumpridas. Para determinar quais atividades entrariam em cada sprint, foi usado primeiramente o critério da prioridade de cada atividade, e secundariamente, o tempo estimado para o término da atividade.

##### 3.1.1 Backlog de atividades

O backlog de atividades é o repositório inicial das atividades, e onde cada atividade foi posta em uma das três categorias de desenvolvimento, App, Webservice e Site Secretaria. Além disso, cada atividade recebeu uma pontuação de 1 a 15, que determinava a prioridade daquela atividade. 1 é a atividade mais prioritária.

##### 3.1.2 Definição dos Sprints

Um sprint é um período previamente acordado pelo time Agile onde o sistema é de fato implementado. Este período pode variar de acordo com cada equipe, mesmo em equipes dentro da mesma organização ou empresa.

Para o desenvolvimento do sistema de gerenciamento dos protocolos e do aplicativo Android, cada sprint teve duração de 14 dias, onde foram escolhidos várias atividades de acordo com os critérios da sua prioridade de implementação e sua facilidade de implementação dada a situação que o projeto se encontrava.

Tabela 1 – Tabela de definição das atividades.

8 - Tela Principal 9 - Incluir protocolos anônimos  10 - Cadastro de usuário 11 - Incluir protocolos cadastrados 12 - Gravar off-line protocolos para envio posterior 15 - Listar protocolos	7 - Incluir protocolos 13 - Consultar protocolos 14 - Listar protocolos	1 - Criar Banco de Dados 2 - Página inicial  3 - Cadastro de usuário 4 - Controle de acesso  5 - Visualizar protocolos  6 - Mudar status protocolos
--	---	---

**Sprint I:** O primeiro sprint foi feito a modelagem e a implementação do banco de dados, assim como o layout e a wireframe do ambiente web.

**Sprint II:** Nesta segunda etapa foi feita a criação do layout e padronização da interface do ambiente web. Também foram implementados os serviços de criação e inclusão dos protocolos.

**Sprint III:** Implementação das telas do aplicativo Android, e do fluxo principal de telas, também como a funcionalidade de ativar a câmera do aparelho, também foi implementado uma funcionalidade que busca automaticamente a localização do aparelho que está enviando o reporte

**Sprint IV:** Este sprint foi focado na implementação das funcionalidades do aplicativo como o cadastro do usuário, tela de configurações e status dos protocolos enviados

**Sprint V:** No último sprint foram acertado os últimos detalhes do aplicativo, como gravação dos protocolos off-line para envio posterior, busca do status do protocolo e foi finalizado o design da aplicação.

### 3.1.3 Apresentação para a Secretaria de Obras

Após a finalização da primeira versão do aplicativo e do ambiente web do projeto “Cidade Iluminada”, fomos convidados a apresentar nossa solução para o Secretário de Obras de São José dos Campos, o senhor Rene Mina Vernice e sua equipe, na data de 14 de Maio de 2015.

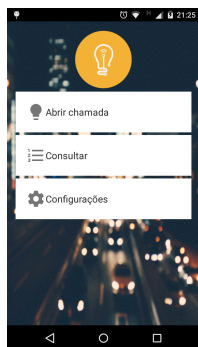
Inicialmente, a ideia do aplicativo não foi bem aceita, pois foi alegado que isso geraria ainda mais trabalho para os funcionários responsáveis por recolher as denúncias relacionadas a iluminação pública. Apesar disso, o conceito do ambiente web para gerenciamento foi bem recebido, pois eles enfrentam muitas dificuldades administrando os chamados já existente.

### 3.1.4 Fluxo de utilização do aplicativo

A seguir será descrito o fluxo principal de utilização do aplicativo Android para os municípios.

Para enviar uma solicitação pelo aplicativo, o usuário deve iniciar a aplicação (Figura 1), e clicar em Abrir Chamado, na próxima tela (Figura 2) serão solicitados alguns dados sobre a chamada.

Figura 1 – Tela inicial



Fonte: Produzido pelos autores

Figura 2 – Tela de nova chamada



Fonte: Produzido pelos autores

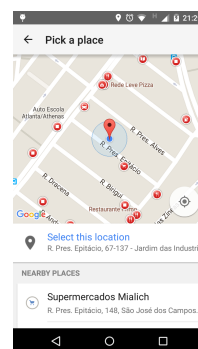
Primeiramente, deve-se tirar uma foto do poste defeituoso (Figura 3) e em seguida selecionar a localização na tela de seleção de endereço via GPS (Figura 4), assim o endereço é preenchido automaticamente (Figura 5).

Figura 3 – Tela de chamada com a foto do poste



Fonte: Produzido pelos autores

Figura 4 – Tela de seleção de endereço via GPS



Fonte: Produzido pelos autores

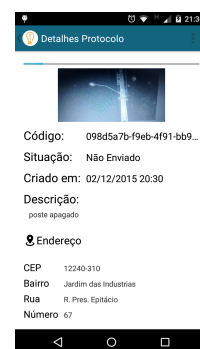
Após a coleta dos dados, o usuário deve clicar em enviar protocolo e logo a seguir, o aplicativo irá iniciar a transferência dos dados para o servidor (Figura 6).

Figura 5 – Tela de chamada com o endereço preenchido automaticamente



Fonte: Produzido pelos autores

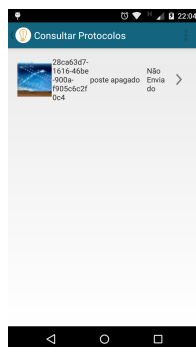
Figura 6 – Tela de envio de protocolo com o envio em progresso



Fonte: Produzido pelos autores

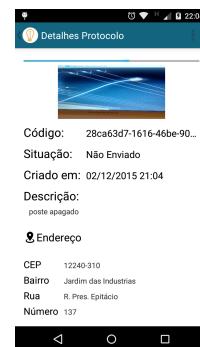
Após enviado o usuário poderá escolher a opção de listar os protocolos enviados (Figura 7) e checar o status do protocolo enviado (Figura 8).

Figura 7 – Tela de lista de protocolos feitos



Fonte: Produzido pelos autores

Figura 8 – Tela de detalhes de protocolo



Fonte: Produzido pelos autores

O usuário também pode configurar a aplicação com o CEP padrão, e caso deseje mandar chamados em modo anônimo. Neste caso, o usuário deverá informar seu nome e e-mail para contato (Figura 9).

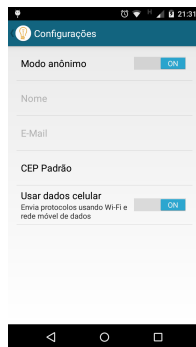
## 3.2 Sistema de gerenciamento de pontos de iluminação de ordens de serviço

Como segunda fase do trabalho, foi definido que seria desenvolvido um ambiente onde os funcionários da Secretaria de Obras pudessem fazer o gerenciamento de protocolos e ordens de serviço.

### 3.2.1 Entrevista inicial ao cliente

Anteriormente a primeira entrevista, foi realizada uma apresentação do conceito do projeto para o Secretário de Obras Renê Vernice e outros integrantes da equipe. O conceito do projeto foi não bem aceito por todos, pois a Secretaria de Obras não atenderia os chamados abertos pelo aplicativo. Foi acordado que o projeto seguiria, caso fossem feitas

Figura 9 – Tela de configuração de preferências



Fonte: Produzido pelos autores

algumas modificações que ajudassem na organização dos serviços realizados nos pontos de iluminação.

Posteriormente, foi feita uma reunião com a assistente do Secretário, a senhora Cintia Firmino para entendermos melhor o funcionamento do processo atual.

**1. Como é o processo atual?**

Processo de atualização feito a partir da coleta manual do sistema 156, e colocado numa planilha para o tratamento de pedidos duplicados e separação dos casos por zona e bairro.

**2. Como vocês recebem as chamadas dos postes com defeitos?**

Os chamados vêm do sistema 156, ou por memorando de Vereadores que tem prioridade, não possui um canal personalizado apenas para eles vem do sistema 156 (156 possui resposta programada)

**3. Como é feita a ordem de serviço?**

A ordem de serviço é feita pelo Excel, separando os chamados por zona, bairro e rua. E após isso esta planilha é enviada para a empresa URBAM por e-mail.

**4. Vocês acham que é o melhor processo? Como poderia melhorar?**

O processo poderia melhorar se os protocolos já chegassem a nós separados pela região, sem duplicidade, com a data, sem a necessidade de analisar todos os pedidos que vem do 156

**5. Qual seria o processo ideal?**

Receber os protocolos separados pela região, bairro e rua, sem duplicidade. Responder ao município automaticamente.

**6. Vocês separam os postes por tipos? Quais tipos?**

Temos dois tipos, a iluminação ornamental, e a iluminação de vias.

**7. Como vocês agrupam postes da mesma rua? Vocês têm uma base de dados de endereços?**



Os postes são agrupados em até 10 números da mesma rua. Há problemas também em verificar os endereços das praças e vielas pois tem mais de um endereço, além de algumas praças e vielas não constarem na pesquisa do Google. Isso afeta também a troca de lâmpadas ornamentais.

8. **Vocês agrupam as situações das pendências (resolvida, não resolvida, enviada para serviço, etc.)? Se não, como seria o ideal?**

Sim, por planilha, quando uma ordem de serviço é mandada, os pedidos são retirados de uma planilha e colocadas em outra sinalizando que foram mandados e estamos esperando a resposta da URBAM.

9. **Vocês agrupam os tipos de serviços das pendências (troca de lâmpada, troca de reator, etc.)? Se não, como seria o ideal?**

Existem 3 tipos de serviços que nós junto com a URBAM, a troca de lâmpada, a troca de reator, e a troca da fotocélula.

O que será trocado é analisado um a um, pela URBAM que depois reporta o que foi trocado.

10. **Como é feita a identificação de cada poste dentro de uma rua? Pelo número aproximado da casa?**

Ainda não existe um sistema de identificação no momento.

11. **Existe algum código para identificar os postes? Se não, o código ideal deve informar o quê?**

Não existe, é pelo número da casa mais próxima ao poste.

12. **Vocês atualizam informações sobre cada ocorrência após o retorno da ordem de serviço? Se sim, o que vocês atualizam?**

Sim, a URBAM retorna uma planilha com as tarefas executadas e o que foi trocado, e após, estas informações são colocadas numa planilha chamada “Executados”, e depois é respondido para o munícipe no 156.

### 3.2.2 Requisitos do sistema

Quando pensamos em qualidade de software, o foco está em entender os requisitos estabelecidos para assim evitar, ou minimizar a insatisfação do cliente final. Segundo Carvalho, Tavares “A demanda por qualidade tem estimulado a comunidade de software para o desenvolvimento de modelos que conduzam a qualidade dos sistemas. Existe uma forte ligação entre requisitos e qualidade.” (TAVARES, 2006)

Em um sistema computacional, os requisitos definem o escopo e os serviços que o sistema deve oferecer, assim como as restrições que são aplicáveis a suas operações.

De acordo com as respostas levantadas na reunião com a Cíntia, foram então levantados os seguintes requisitos funcionais:

**RF1:** O sistema deverá fazer a coleta automática dos protocolos no sistema do 156 direcionados para a secretaria de obras.

**RF2:** O sistema deverá fazer essa coleta e, de acordo com a informação de cada protocolo, separar em bairros e regiões corretas para cada endereço.

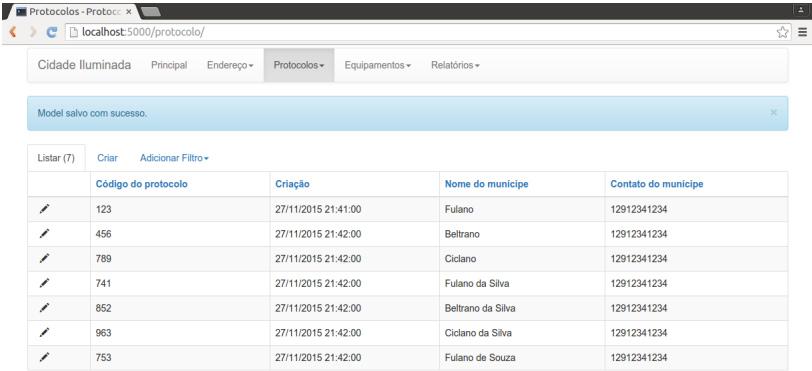
- RF3:** O sistema deverá fazer a coleta sem adicionar protocolos em duplicidade. Dois ou mais protocolos em duplicidade se caracterizam por se referirem ao mesmo ponto de iluminação pública, tendo em vista que os postes são agrupados logicamente de 10 em 10 metros na rua. Por exemplo, se um protocolo se referir a um poste no número 10 da rua, e outro protocolo se referir a um poste no número 12 da rua, isso seria o mesmo poste.
- RF4:** O sistema deverá tentar ligar a cada protocolo a um poste, caso isso falhe, os sistemas devem permitir que o usuário faça essa correção manualmente.
- RF5:** O sistema deverá gerar a ordem de serviço em formato de planilha com, no máximo 50 pendências. Na geração da ordem de serviço, o sistema deverá dar preferência em agrupar protocolos do mesmo bairro e da mesma região da cidade.
- RF6:** O sistema deverá permitir a atualização da situação de execução e qual foi o tipo de manutenção realizada de cada protocolo das ordens de serviço.
- RF7:** O sistema deverá identificar as ordens de serviço que foram executadas para que o munícipe seja alertado.
- RF8:** O sistema deverá permitir que se faça buscas na base de postes e protocolos.

### 3.2.3 Fluxo de utilização do sistema de gerenciamento

A seguir será descrito o fluxo principal de utilização do sistema de gerenciamento destinado à Secretaria Municipal de Obras de São José dos Campos.

Para iniciar, o usuário faz o login no sistema e abre o menu Protocolos onde os protocolos estão listados, como mostrado na Figura 10.

Figura 10 – Lista de protocolos



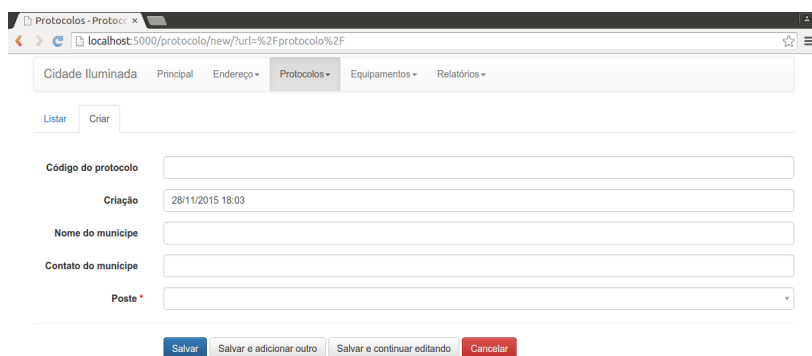
Listar (7)	Criar	Adicionar Filtro			
	Código do protocolo	Criação	Nome do municípe	Contato do municípe	
	123	27/11/2015 21:41:00	Fulano	12912341234	
	456	27/11/2015 21:42:00	Beltrano	12912341234	
	789	27/11/2015 21:42:00	Ciclano	12912341234	
	741	27/11/2015 21:42:00	Fulano da Silva	12912341234	
	852	27/11/2015 21:42:00	Beltrano da Silva	12912341234	
	963	27/11/2015 21:42:00	Ciclano da Silva	12912341234	
	753	27/11/2015 21:42:00	Fulano de Souza	12912341234	

Fonte: Produzido pelos autores

Para criar um novo protocolo, deve se clicar em Criar, onde será aberto o formulário de novo protocolo, como mostrado na Figura 11. Note as informações necessárias para se poder criar um novo protocolo, sendo a informação mais importante qual poste que o protocolo se refere (Figura 12). Após a inserção, o protocolo será mostrado na lista inicial.

Para editar um protocolo, na tela da lista de protocolos (Figura 10) deve-se clicar no ícone em forma de lápis, no canto esquerdo da linha do protocolo que o usuário deseja editar, e será aberto o formulário idêntico ao formulário de criação, com os dados do protocolo selecionado disponíveis para edição.

Figura 11 – Novo protocolo



Protocolos - Protocolo

localhost:5000/protocolo/new/?url=%2Fprotocolo%2F

Cidade Iluminada Principal Endereço Protocolos Equipamentos Relatórios

Listar Criar

Código do protocolo

Criação 28/11/2015 18:03

Nome do municípe

Contato do municípe

Poste \*

Salvar Salvar e adicionar outro Salvar e continuar editando Cancelar

Fonte: Produzido pelos autores

Figura 12 – Selecionar postes

Protocolos - Protocolos

localhost:5000/protocolo/new?url=%2Fprotocolo%2F

Cidade Iluminada Principal Endereço Protocolos Equipamentos Relatórios

Listar Criar

Código do protocolo 95621

Criação 28/11/2015 18:03

Nome do município Arthur de Paula Bressan

Contato do município 12912341234

Poste \*

122274

Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 20

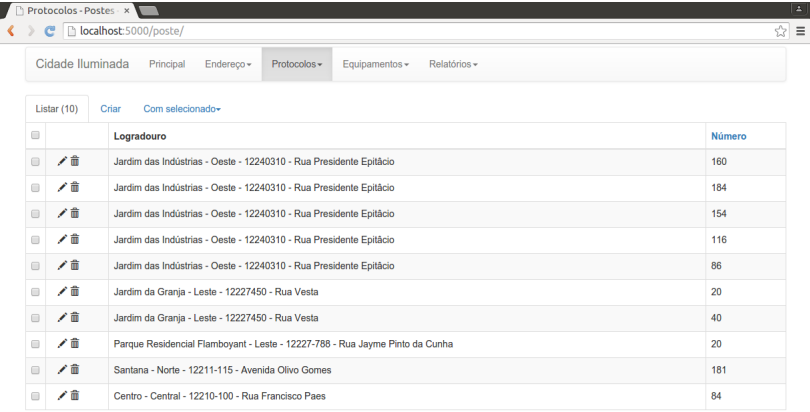
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 40

Fonte: Produzido pelos autores











Ao associar o protocolo a um poste, essa associação pode ser confirmada através da lista de postes (Figura 13).

Ao selecionar o poste que se deseja ver em detalhes, é mostrado todo o seu histórico de manutenção, como mostrado na Figura 14. A partir dessa visão, para cada item de manutenção do poste, pode estar associado um ou vários protocolos, como por exemplo na Figura 15.

Figura 13 – Lista de postes



The screenshot shows a web browser window with the address bar displaying 'localhost:5000/poste/'. The application has a navigation bar with tabs: 'Cidade Iluminada', 'Principal', 'Endereço', 'Protocolos', 'Equipamentos', and 'Relatórios'. The 'Protocolos' tab is active. Below the navigation bar, there is a section with 'Listar (10)', 'Criar', and 'Com selecionado'. The main content is a table with 10 rows of data. Each row has a checkbox, an edit/delete icon, a location and address, and a number. The locations include 'Logradouro', 'Jardim das Indústrias - Oeste', 'Jardim da Granja - Leste', 'Parque Residencial Flamboyant', 'Santana - Norte', and 'Centro - Central'.

			Número
<input type="checkbox"/>		Logradouro	
<input type="checkbox"/>		Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio	160
<input type="checkbox"/>		Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio	184
<input type="checkbox"/>		Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio	154
<input type="checkbox"/>		Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio	116
<input type="checkbox"/>		Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio	86
<input type="checkbox"/>		Jardim da Granja - Leste - 12227450 - Rua Vesta	20
<input type="checkbox"/>		Jardim da Granja - Leste - 12227450 - Rua Vesta	40
<input type="checkbox"/>		Parque Residencial Flamboyant - Leste - 12227-788 - Rua Jayme Pinto da Cunha	20
<input type="checkbox"/>		Santana - Norte - 12211-115 - Avenida Olivo Gomes	181
<input type="checkbox"/>		Centro - Central - 12210-100 - Rua Francisco Paes	84

Fonte: Produzido pelos autores

Figura 14 – Histórico de manutenção do poste

Cidade Iluminada Principal Endereço Protocolos Equipamentos Relatórios

Listar Criar Editar

Logradouro Jardim da Granja - Leste - 12227450 - Rua Vesta

Número 20

Salvar Salvar e adicionar outro Salvar e continuar editando Cancelar

### Histórico de manutenção

Criação	Status	Protocolos da pendência
27/11/2015 21:42:16	aberto	<a href="#">Ver protocolos (2)</a>

Fonte: Produzido pelos autores

Figura 15 – Protocolos associados ao poste

Cidade Iluminada Principal Endereço Protocolos Equipamentos Relatórios

Listar (2) Criar Adicionar Filtro

\* Item Manutencao / Id igual 53 Limpar Filtros

	Código do protocolo	Criação	Nome do municípe	Contato do municípe
	963	27/11/2015 21:42:00	Ciclano da Silva	12912341234
	95621	28/11/2015 18:03:00	Arthur de Paula Bressan	12912341234

Fonte: Produzido pelos autores



Para o usuário da Secretaria de Obras, no menu Principal é possível visualizar as Ordens de Serviço em aberto e também gerar novas Ordens de Serviço, como pode ser visto na Figura 16.

Para gerar uma nova Ordem de Serviço, o usuário deverá clicar na região desejada, selecionar os bairros (Figura 17) e, por fim, confirmar os postes com pendências, como mostrado na Figura 18.

Após selecionar todos os postes com pendências que se deseja adicionar na nova Ordem de Serviço, a Ordem de Serviço pode então ser finalmente criada ao clicar no botão de Gerar Ordem de Serviço. Assim uma nova Ordem de Serviço é gerada como mostrado na Figura 19.

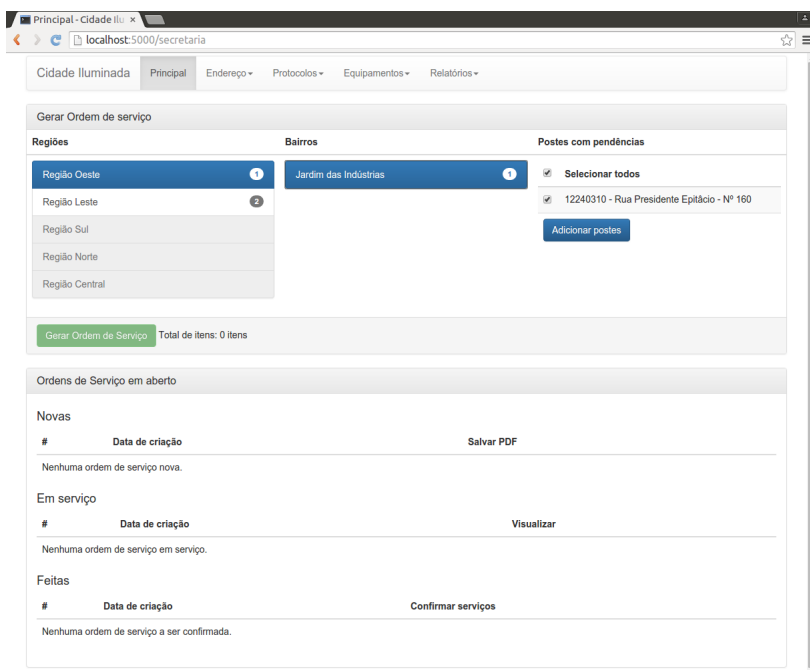
Figura 16 – Tela principal da secretaria

A imagem mostra a interface web da Secretaria de Obras. No topo, há uma barra de navegação com o menu "Principal" selecionado. Abaixo, há uma seção "Gerar Ordem de serviço" com três colunas: "Regiões", "Bairros" e "Postes com pendências". Na coluna "Regiões", há uma lista com "Região Oeste", "Região Leste", "Região Sul", "Região Norte" e "Região Central". Na coluna "Postes com pendências", há um botão "Selecionar todos" e um botão "Adicionar postes". Abaixo disso, há um botão "Gerar Ordem de Serviço" e o texto "Total de itens: 0 itens".

Abaixo da seção "Gerar Ordem de serviço", há uma seção "Ordens de Serviço em aberto" com três sub-seções: "Novas", "Em serviço" e "Feitas". Cada sub-seção tem uma tabela com colunas para "#", "Data de criação" e uma ação. Na sub-seção "Novas", a ação é "Salvar PDF". Na sub-seção "Em serviço", a ação é "Visualizar". Na sub-seção "Feitas", a ação é "Confirmar serviços".

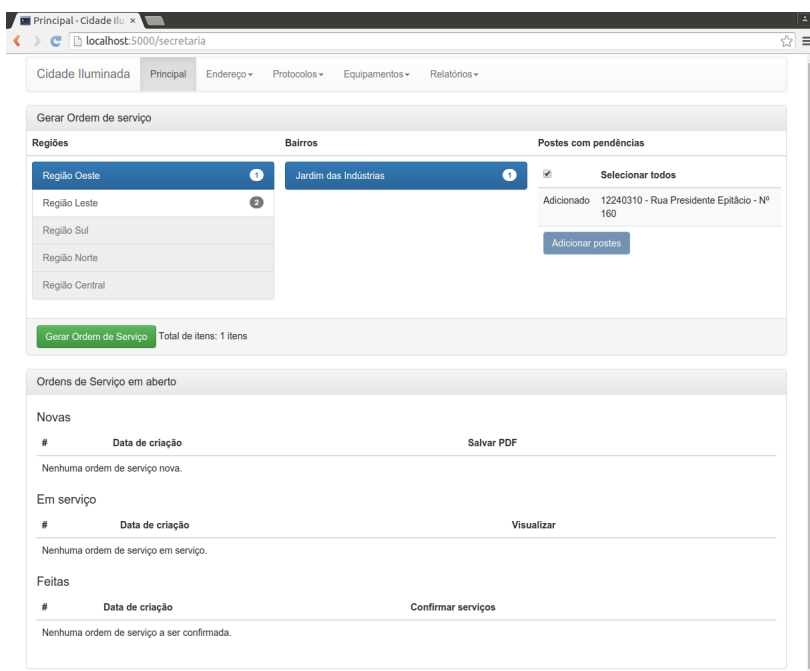
Fonte: Produzido pelos autores

Figura 17 – Tela principal da secretaria com a região e bairro selecionado



Fonte: Produzido pelos autores

Figura 18 – Postes da região oeste selecionados



Fonte: Produzido pelos autores

Figura 19 – Ordem de Serviço gerada

The screenshot shows a web application interface for generating service orders. The browser address bar indicates the URL is localhost:5000/secretaria. The application has a navigation menu with options: Cidade Iluminada, Principal, Endereço, Protocolos, Equipamentos, and Relatórios. The main section is titled 'Gerar Ordem de serviço' and contains three columns: Regiões, Bairros, and Postes com pendências. The Regiões column lists five regions: Região Oeste, Região Leste, Região Sul, Região Norte, and Região Central. The Postes com pendências column has a checkbox labeled 'Selecionar todos' and a button labeled 'Adicionar postes'. Below this section is a green button labeled 'Gerar Ordem de Serviço' and a text label 'Total de itens: 0 itens'. The bottom section is titled 'Ordens de Serviço em aberto' and contains three sub-sections: 'Novas', 'Em serviço', and 'Feitas'. The 'Novas' section has a table with columns '#', 'Data de criação', and 'Salvar PDF'. It contains one row with the number 30 and the date 28/11/2015 18:10:57, with a 'Salvar' button. The 'Em serviço' section has a table with columns '#', 'Data de criação', and 'Visualizar', and contains no data. The 'Feitas' section has a table with columns '#', 'Data de criação', and 'Confirmar serviços', and contains no data.

Regiões	Bairros	Postes com pendências
Região Oeste		<input checked="" type="checkbox"/> Selecionar todos
Região Leste		<button>Adicionar postes</button>
Região Sul		
Região Norte		
Região Central		

Gerar Ordem de Serviço Total de itens: 0 itens

### Ordens de Serviço em aberto

#### Novas

#	Data de criação	Salvar PDF
30	28/11/2015 18:10:57	<button>Salvar</button>

#### Em serviço

#	Data de criação	Visualizar
Nenhuma ordem de serviço em serviço.		

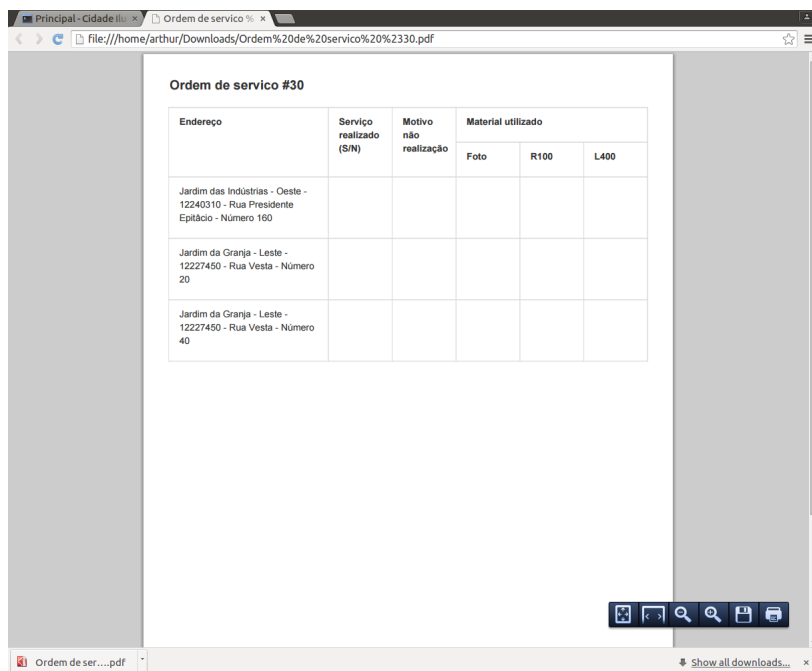
#### Feitas

#	Data de criação	Confirmar serviços
Nenhuma ordem de serviço a ser confirmada.		

Fonte: Produzido pelos autores

Tanto o usuário da Secretaria de Obras quanto o usuário da URBAM tem acesso ao PDF que será gerado, como mostrado na Figura 20. A finalidade do PDF é servir como apoio em campo para os operários da URBAM para melhor relatarem os serviços feitos e equipamentos utilizados. Esse PDF mostra uma grade com o endereço completo dos postes da Ordem de Serviço, bem como os equipamentos previamente cadastrados que podem ser utilizados. Além disso, esta grade possui um campo para o preenchimento do motivo de não realização do serviço, caso seja necessário.

Figura 20 – Lista de serviços da Ordem de Serviço



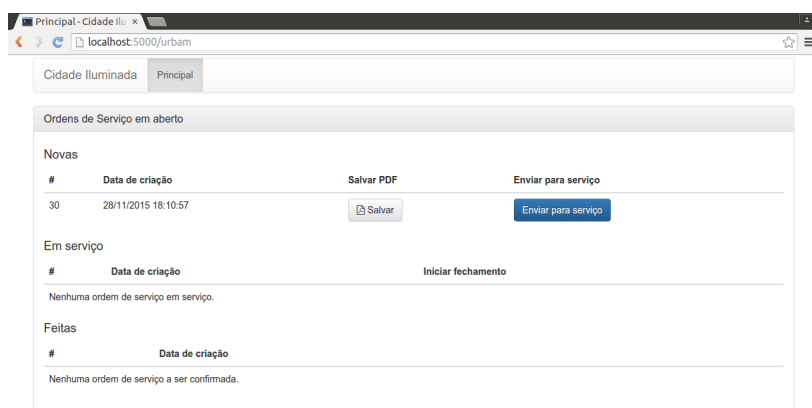
The image shows a PDF document titled "Ordem de serviço #30". The document contains a table with the following structure:

Endereço	Serviço realizado (S/N)	Motivo não realização	Material utilizado		
			Foto	R100	L400
Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio - Número 160					
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 20					
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 40					

Fonte: Produzido pelos autores

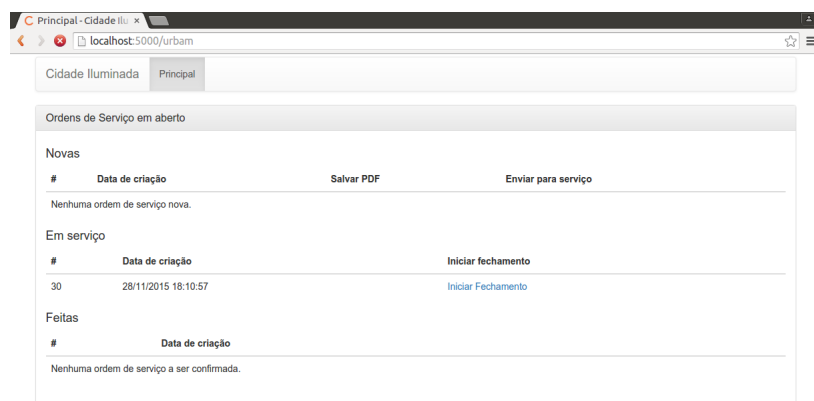
Assim que uma equipe de manutenção for despachada, o usuário da URBAM (Figura 21) deverá sinalizar este envio clicando no botão Enviar para serviço. Após isso, a Ordem de Serviço irá mudar de status para Em serviço, e estará disponível para que seja realizado o seu eventual fechamento (Figura 22).

Figura 21 – Tela principal do usuário da URBAM



Fonte: Produzido pelos autores

Figura 22 – Ordem de Serviço sinalizada como Em serviço



Cidade Iluminada Principal

Ordens de Serviço em aberto

**Novas**

#	Data de criação	Salvar PDF	Enviar para serviço
Nenhuma ordem de serviço nova.			

**Em serviço**

#	Data de criação	Iniciar fechamento
30	28/11/2015 18:10:57	<a href="#">Iniciar Fechamento</a>

**Feitas**

#	Data de criação
Nenhuma ordem de serviço a ser confirmada.	

Fonte: Produzido pelos autores

Para fechar uma Ordem de Serviço em aberto, o usuário da URBAM deverá clicar em Iniciar Fechamento, e então o usuário será redirecionado para a tela de fechamento (Figura 23). Neste momento, etapa o usuário irá informar ao sistema se o serviço foi realizado, e o material utilizado, como a Figura 24 mostra. Após o fechamento de todos os serviços, o status da Ordem de Serviço ficará como Feita, ilustrado na Figura 25.

Figura 23 – Tela de fechamento de serviço da URBAM

Poste	Realizado	Material utilizado			Motivo não realização	Salvar serviço
		Foto	R100	L400		
Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio - Número 160	<input type="radio"/> Sim <input type="radio"/> Não	0	0	0		Salvar >
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 20	<input type="radio"/> Sim <input type="radio"/> Não	0	0	0		Salvar >
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 40	<input type="radio"/> Sim <input type="radio"/> Não	0	0	0		Salvar >

Fonte: Produzido pelos autores



Figura 24 – Tela de fechamento preenchida

Poste	Realizado	Material utilizado			Motivo não realização	Salvar serviço
		Foto	R100	L400		
Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio - Número 160	<input checked="" type="radio"/> Sim <input type="radio"/> Não	3	3	3		Salvar >
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 20	<input type="radio"/> Sim <input checked="" type="radio"/> Não	0	0	0	Não encontrado	Salvar >
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 40	<input type="radio"/> Sim <input checked="" type="radio"/> Não	0	0	0	Já realizado	Salvar >

Fonte: Produzido pelos autores

Figura 25 – Ordem de Serviço feita

Poste	Realizado	Material utilizado			Motivo não realização	Salvar serviço
		Foto	R100	L400		
Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio - Número 160	✓	3	3	3		
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 20	✗				Não encontrado	
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 40	✗				Já realizado	

Fonte: Produzido pelos autores

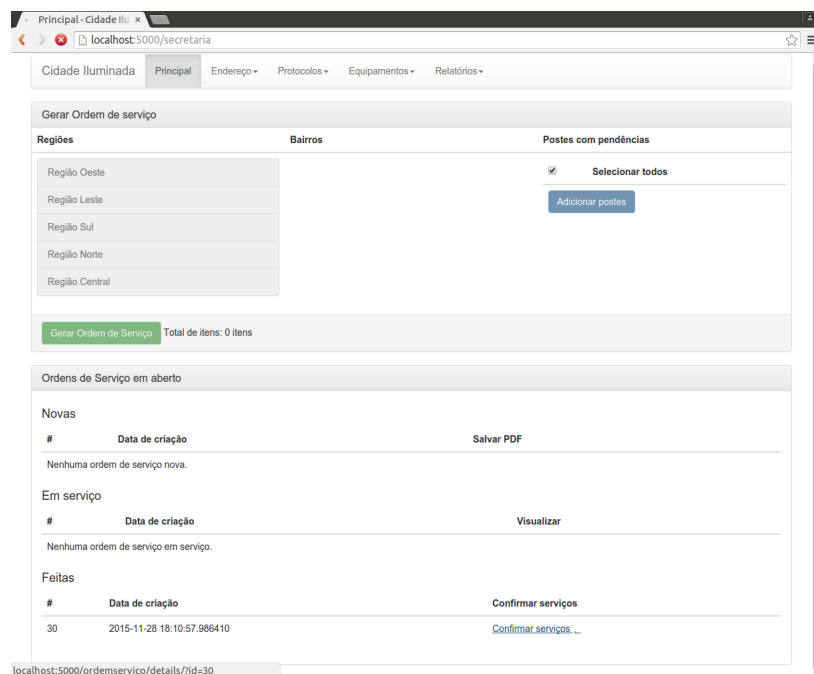
Com a Ordem de Serviço Feita, o usuário da Secretaria deve confirmar o material utilizado em cada serviço realizado, além de conferir o motivo de alguns dos serviços não terem sido executados.

Para confirmar uma Ordem de Serviço, o usuário da Secretaria deve abrir a página principal e procurar na lista de Ordens de Serviço Feitas a Ordem de Serviço que se deseja confirmar e clicar no link de Confirmar Serviços (Figura 26).

Após isso, irá ser aberta uma tela similar à tela de fechamento de serviço da URBAM. Porém, para cada serviço irá ser mostrado o material utilizado ou o motivo de não realização do serviço (Figura 27), um campo para o comentário por parte da Secretaria e um botão para confirmar o serviço.

Assim que a Ordem de Serviço tiver com todos os serviços confirmados, a sua situação irá mudar para Feita, o que sinaliza que não é necessário mais nenhuma ação (Figura 28).

Figura 26 – Lista de Ordem de Serviço feitas



Fonte: Produzido pelos autores

Figura 27 – Confirmação da Ordem de Serviço

Poste	Realizado	Material utilizado			Motivo não realização	Protocolos	Comentários	Confirmar serviço
		Foto	R100	L400				
Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio - Número 160	✓	3	3	3		<a href="#">Ver protocolos (1)</a>	<input type="text"/>	<a href="#">Confirmar &gt;</a>
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 20	✗				Não encontrado	<a href="#">Ver protocolos (2)</a>	<input type="text"/>	<a href="#">Confirmar &gt;</a>
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 40	✗				Já realizado	<a href="#">Ver protocolos (1)</a>	<input type="text"/>	<a href="#">Confirmar &gt;</a>

Fonte: Produzido pelos autores

Figura 28 – Ordem de Serviço confirmada

Poste	Realizado	Material utilizado			Motivo não realização	Protocolos	Comentários	Confirmar serviço
		Foto	R100	L400				
Jardim das Indústrias - Oeste - 12240310 - Rua Presidente Epitácio - Número 160	✓	3	3	3		<a href="#">Ver protocolos (1)</a>		
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 20	✗				Não encontrado	<a href="#">Ver protocolos (2)</a>	Pedir mais detalhes	
Jardim da Granja - Leste - 12227450 - Rua Vesta - Número 40	✗				Já realizado	<a href="#">Ver protocolos (1)</a>	Município informado	

Fonte: Produzido pelos autores

## Considerações finais

A gestão de pendências relativas a iluminação pública da prefeitura de São José dos Campos é burocrática, o que reflete na lentidão do atendimento e manutenção de um serviço tão essencial ao bem estar e segurança da população. O fato da responsabilidade pela manutenção ter sido passada para as prefeituras recentemente também é um fator crítico no desempenho do atendimento, já que as prefeituras tiveram que se adaptar rapidamente.

Partindo desta problemática, incentivados pela ideia de agilizar este processo, desenvolvemos o projeto Cidade Iluminada, onde o ambiente web, que passou pela validação da equipe da Secretaria de Obras da Prefeitura Municipal de São José dos Campos provou reduzir a carga de trabalho e facilitar o recolhimento e análise das denúncias recebidas, automatizando um trabalho que era feito manualmente.

Infelizmente o aplicativo ainda não será disponibilizado para os munícipes a pedido do cliente, que alega que este canal de denúncias poderia neste primeiro momento congestionar os recebimentos de denúncias já que a equipe ainda não possui um processo estruturado para a demanda, o aplicativo será usado por funcionários para a identificação de pontos de iluminação defeituosos.

Como trabalho futuro, foi proposto que o aplicativo seja liberado para o uso dos munícipes, integrando ao ambiente web para que ele receba as denúncias diretamente. Para que o aplicativo seja liberado, também é necessário que além de Android seja desenvolvido para iOS assim alcançando todos os públicos.

Relacionado ao trabalho em questão, seria interessante que futuramente novas funcionalidades sejam integradas, utilizando o aplicativo para outras utilizações além de denúncias relacionadas a iluminação pública.

Uma outra possibilidade é a melhoria da interface gráfica, tornando o ambiente web mais amigável para o uso de funcionários com vários níveis de conhecimento.

# Public lighting maintenance management system

## CidadeIluminada

Arthur de Paula Bressan  
(IBTA)

Thais Matias Nogueira  
(IBTA)

2015

### Abstract

This paper presents a study on mobile development, explaining the implementation of a software prototype that will assist in receiving complaints and analysis related to street lighting in the city of São José dos Campos. The application was developed on Android and will be responsible for sending these complaints to a web environment. The web application will be responsible for responding to requests from the Android application and process them so that the maintenance of public lighting points is done in a faster and less bureaucratic way.

**Keywords:** Public lighting system. Android Application. Web system for maintenance.

### Referências

ALEMBIC. *Alembic Documentation Website*. 2010. Website. Disponível em: <<https://alembic.readthedocs.org/en/latest/>>. Acesso em: 06 nov 2015. Citado na página 6.

ALLIANCE, S. *Scrum Values*. 2015. Website. Disponível em: <[https://www.scrumalliance.org/why-scrum/core-scrum-values-roles?gclid=CjwKEAjwpaqvBRCxzIGoxs6v2TkSJADel\\_MIzZiwKysiX\\_\\_QU25cWumb4QQi5ae356H3APHD\\_R79FNRoCJaXw\\_wcB](https://www.scrumalliance.org/why-scrum/core-scrum-values-roles?gclid=CjwKEAjwpaqvBRCxzIGoxs6v2TkSJADel_MIzZiwKysiX__QU25cWumb4QQi5ae356H3APHD_R79FNRoCJaXw_wcB)>. Acesso em: 12 out 2015. Citado na página 4.

AMADOR, J. G. *Estudo aponta aumento na penetração de smartphones na América Latina*. 2015. Notícia. Disponível em: <[http://www.correiobraziliense.com.br/app/noticia/tecnologia/2015/03/12/interna\\_tecnologia,475253/estudo-aponta-aumento-na-penetracao-de-smartphones-na-america-latina.shtml](http://www.correiobraziliense.com.br/app/noticia/tecnologia/2015/03/12/interna_tecnologia,475253/estudo-aponta-aumento-na-penetracao-de-smartphones-na-america-latina.shtml)>. Acesso em: 06 nov 2015. Citado na página 4.

ANEEL, A. N. de E. E. *Resolução Normativa 414/2010: atualizada até a REN 499/2012*. 2015. Artigo. Disponível em: <[http://www.aneel.gov.br/biblioteca/downloads/livros/REN\\_414\\_2010\\_atual\\_REN\\_499\\_2012.pdf](http://www.aneel.gov.br/biblioteca/downloads/livros/REN_414_2010_atual_REN_499_2012.pdf)>. Acesso em: 12 abr 2015. Citado na página 1.

ANGULARJS. *AngularJS FAQ*. 2006. FAQ. Disponível em: <<https://docs.angularjs.org/misc/faq>>. Acesso em: 06 nov 2015. Citado na página 9.

AVER, A. *A relação Iluminação Pública e Criminalidade*. 2015. Especialize Revista Online-IPOG. Disponível em: <[www.ipog.edu.br/download-arquivo-site.sp?arquivo=a\\_relacao\\_iluminacao\\_publica\\_e\\_criminalidade\\_152947.pdf](http://www.ipog.edu.br/download-arquivo-site.sp?arquivo=a_relacao_iluminacao_publica_e_criminalidade_152947.pdf)>. Acesso em: 12 mai 2015. Citado na página 1.

BREITMAN, K. K. *EVOLUÇÃO DE CENÁRIOS*. Tese (Doutorado) — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Rio de Janeiro, 2000. Disponível em: <<http://www.loa.istc.cnr.it/Guizzardi/SELMAS-CR.pdf>>. Acesso em: 12 abr 2015. Citado na página 3.

COMTECH. *Android Switchers Drive iOS Growth in Europe's Big Five*. 2010. Website. Disponível em: <<http://www.kantarworldpanel.com/global/News/Android-Switchers-Drive-iOS-Growth-in-Europes-Big-Five-Countries>>. Acesso em: 06 nov 2015. Citado na página 8.

CROCKFORD, D. *JSON: The Fat-Free Alternative to XML*. 2006. Website. Disponível em: <<http://www.json.org/fatfree.html>>. Acesso em: 06 nov 2015. Citado na página 9.

EMARKETER. *Nearly 400 Million in Latin America Used Mobile Phones in 2014*. 2015. Notícia. Disponível em: <<http://www.emarketer.com/Article/Nearly-400-Million-Latin-America-Used-Mobile-Phones-2014/1011818>>. Acesso em: 06 nov 2015. Citado na página 4.

FABRIC. *Fabric Documentation Website*. 2010. Website. Disponível em: <<http://www.fabfile.org/>>. Acesso em: 06 nov 2015. Citado na página 6.

FABTOOLS. *Fabtools Documentation Website*. 2010. Website. Disponível em: <<http://fabtools.readthedocs.org/en/latest/>>. Acesso em: 06 nov 2015. Citado na página 7.

FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Tese (Doutorado) — Information and Computer Science, University Of California, Irvine, Irvine, CA 92697, United States, 2000. Disponível em: <[http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)>. Acesso em: 06 nov 2015. Citado na página 10.

FLASK. *Flask Website*. 2010. Website. Disponível em: <<http://flask.pocoo.org/docs/0.10/>>. Acesso em: 06 nov 2015. Citado na página 5.

FLASK-ADMIN. *Flask-Admin Documentation Website*. 2010. Website. Disponível em: <<https://flask-admin.readthedocs.org/en/latest/>>. Acesso em: 06 nov 2015. Citado na página 6.

FLASK-MIGRATE. *Flask-Migrate Documentation Website*. 2010. Website. Disponível em: <<https://flask-migrate.readthedocs.org/en/latest/>>. Acesso em: 06 nov 2015. Citado na página 6.

FLASK-SCRIPT. *Flask-Script Documentation Website*. 2010. Website. Disponível em: <<https://flask-script.readthedocs.org/en/latest/>>. Acesso em: 06 nov 2015. Citado na página 6.

FLASK-SECURITY. *Flask-Security Documentation Website*. 2010. Website. Disponível em: <<https://pythonhosted.org/Flask-Security/>>. Acesso em: 06 nov 2015. Citado na página 7.

FLASK-SQLALCHEMY. *Flask-SQLAlchemy Website*. 2010. Website. Disponível em: <<http://flask-sqlalchemy.pocoo.org/2.1/>>. Acesso em: 06 nov 2015. Citado na página 5.

GSON. *GSON Repository*. 2015. Website. Disponível em: <<https://github.com/google/gson>>. Acesso em: 06 nov 2015. Citado na página 8.

JODA-TIME. *Joda-Time Home*. 2015. Website. Disponível em: <<http://www.joda.org/joda-time/>>. Acesso em: 06 nov 2015. Citado na página 8.

JQUERY. *JQuery Website*. 2006. Website. Disponível em: <<https://jquery.com/>>. Acesso em: 06 nov 2015. Citado na página 10.

LIMA, G. *Norma da Aneel sobre iluminação pública prejudica municípios, diz Confederação*. 2015. Notícia. Disponível em: <[http://www2.camara.leg.br/atividade-legislativa/comissoes/comissoes-permanentes/cdu/noticias/norma\\_da\\_aneel\\_sobre\\_iluminacao\\_publica\\_prejudica\\_municipios\\_diz\\_confederacao](http://www2.camara.leg.br/atividade-legislativa/comissoes/comissoes-permanentes/cdu/noticias/norma_da_aneel_sobre_iluminacao_publica_prejudica_municipios_diz_confederacao)>. Acesso em: 12 out 2015. Citado na página 4.

NEWTON, C. *Correlacao Entre Iluminacao Pública e Taxa de Criminalidade*. 2015. Artigo revista online E-How. Disponível em: <[http://www.ehow.com.br/correlacao\\_entre\\_iluminacao\\_publica\\_taxa\\_criminalidade\\_fatos\\_305151/](http://www.ehow.com.br/correlacao_entre_iluminacao_publica_taxa_criminalidade_fatos_305151/)>. Acesso em: 12 mai 2015. Citado na página 1.

NIEMEYER, P. *Learning Java*. 4. ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2013. Citado na página 7.

OBRAS, S. de. *Equipes substituem lâmpadas nas regiões leste e sudeste*. 2015. Notícia. Disponível em: <[http://www.sjc.sp.gov.br/secretarias/obras/noticia.aspx?noticia\\_id=21903](http://www.sjc.sp.gov.br/secretarias/obras/noticia.aspx?noticia_id=21903)>. Acesso em: 12 out 2015. Citado na página 2.

PRESSMAN, R. S. *Engenharia de software: uma abordagem profissional*. 7. ed. Porto Alegre: AMGH, 2011. Citado 2 vezes nas páginas 2 e 3.

PSF, P. S. F. *Python 2.7 license*. 2010. Website. Disponível em: <<https://www.python.org/download/releases/2.7.3/license/>>. Acesso em: 07 set 2015. Citado na página 5.

PYPI, P. P. I. *Python Package Index*. 2010. Website. Disponível em: <<https://pypi.python.org/pypi>>. Acesso em: 07 set 2015. Citado na página 5.

RAUSCHMAYER, A. *Learning Java*. 1. ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2014. Citado na página 9.

RETROFIT. *Retrofit*. 2015. Website. Disponível em: <<http://square.github.io/retrofit/>>. Acesso em: 06 nov 2015. Citado na página 8.



- SQLALCHEMY. *SQLAlchemy Website*. 2010. Website. Disponível em: <<http://www.sqlalchemy.org/features.html>>. Acesso em: 06 nov 2015. Citado na página 5.
- STATISTA. *Market share held by mobile operating systems in Brazil from January 2012 to July 2015*. 2010. Website. Disponível em: <<http://www.statista.com/statistics/262167/market-share-held-by-mobile-operating-systems-in-brazil/>>. Acesso em: 06 nov 2015. Citado na página 8.
- SUGAR. *Sugar Repository*. 2015. Website. Disponível em: <<https://github.com/satyan/sugar>>. Acesso em: 06 nov 2015. Citado na página 9.
- SUTHERLAND, J. *Manifesto for Agile Software Development*. 2015. Website. Disponível em: <<http://www.agilemanifesto.org/>>. Acesso em: 12 out 2015. Citado na página 3.
- TAVARES, H. C. *Visão Geral Sobre Requisitos*. 2006. Revista Tematec - Serviço Federal de Processamento de Dados. Disponível em: <<http://www1.serpro.gov.br/publicacoes/tematec/pubtem60.htm>>. Acesso em: 06 nov 2015. Citado na página 16.
- UNICODECSV. *UnicodeCSV Documentation Website*. 2010. Website. Disponível em: <<http://unicodecsv.readthedocs.org/en/latest/>>. Acesso em: 06 nov 2015. Citado na página 7.
- XHTML2PDF. *XHTML2PDF Repository*. 2010. Website. Disponível em: <<https://github.com/xhtml2pdf/xhtml2pdf>>. Acesso em: 06 nov 2015. Citado na página 6.