



UNIVERSIDADE FEDERAL DO MARANHÃO
Curso de Ciência da Computação

André Luiz Dorneles Pires

**SIVIO - Uma Proposta para Transparência de
Gastos em Obras na Universidade Federal do
Maranhão**

São Luís
2018

André Luiz Dorneles Pires

SIVIO - Uma Proposta para Transparência de Gastos em Obras na Universidade Federal do Maranhão

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof^{fa}. Dr^a. Simara Vieira da Rocha

São Luís

2018

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Pires, André Luiz Dorneles.

SIVIO - Uma Proposta para Transparência de Gastos em Obras na Universidade Federal do Maranhão / André Luiz Dorneles Pires. - 2018.

36 f.

Orientador(a): Simara Vieira da Rocha.

Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, Universidade Federal do Maranhão, São Luís - MA, 2018.

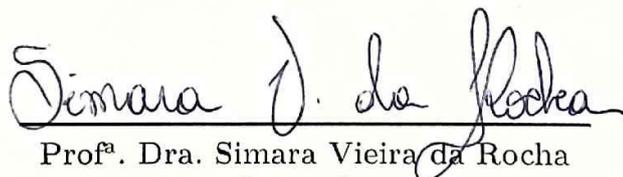
1. Aplicação Móvel. 2. Georreferenciamento. 3. SIVIO. 4. Transparência Governamental. I. Rocha, Simara Vieira da. II. Título.

André Luiz Dorneles Pires

SIVIO - Uma Proposta para Transparência de Gastos em Obras na Universidade Federal do Maranhão

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

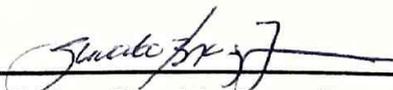
Trabalho aprovado em: São Luís - MA, 12 de Julho de 2018:



Prof^ª. Dra. Simara Vieira da Rocha
Orientador
Universidade Federal do Maranhão



Prof. Msc. Carlos Eduardo Portela
Serra de Castro
Universidade Federal do Maranhão



Prof. Dr. Geraldo Braz Junior
Universidade Federal do Maranhão

São Luís - MA

2018

Agradecimentos

Primeiramente a Deus, o dador da vida.

À minha família, especialmente aos meus pais, Luiz e Maria José, por sempre me apoiarem, acreditarem em mim e estarem dispostos a me ajudar no que for preciso.

À minha namorada, Késia, pelo apoio, paciência e incentivo demonstrados durante a minha trajetória acadêmica.

À professora Simara Vieira da Rocha, pela paciência e suporte, tanto como orientadora, quanto como coordenadora e mestra.

Ao professor Geraldo Braz Junior, por ter aceitado compor a banca, e pelo conhecimento transmitido e interesse pessoal demonstrado nas disciplinas em que fui seu aluno.

Ao professor Carlos Portela, por ter aceitado compor a banca, e pela transmissão e incentivo à procura de conhecimento.

Aos amigos de graduação, Wesley, Guilherme, Bruno, Alynne, Igor, Laudelino, Kelson, Ricardo e Giovanni tanto pelos momentos de descontração, quanto de estudo e de conhecimento compartilhado.

*"Quando a sabedoria entrar no seu coração
E o conhecimento se tornar agradável para a sua alma,
O raciocínio o guardará
E o discernimento o protegerá"*
(Provérbios 2:10, 11)

Resumo

A transparência é um importante componente de qualquer governo democrático, já que é por meio dela que a sociedade pode fiscalizar as ações da administração pública. No Brasil, a Constituição Federal e outras leis específicas, expressam a necessidade de que toda a atividade da Administração Pública seja transparente, inclusive exigindo o uso de tecnologias, como a Internet para alcançar esse objetivo. Visando desenvolver uma forma alternativa à obtenção de informação sobre os gastos em obras na Universidade Federal do Maranhão (UFMA), esse trabalho apresenta o Sistema de Visualização de Obras (SIVIO), uma aplicação móvel que faz uso de georreferenciamento para mostrar a localização dessas obras. O objetivo desta ferramenta é utilizar tecnologias que facilitem a busca e o acesso à informação e assim incentivar os cidadãos, em especial a comunidade acadêmica da UFMA a tomarem mais conhecimento sobre as ações da administração da universidade.

Palavras-chave: SIVIO, Transparência Governamental, Georreferenciamento, Aplicação Móvel.

Abstract

Transparency is an important component of any democratic government, since it is through it that society can oversee the actions of public administration. In Brazil, the Federal Constitution and other specific laws express the need for all Public Administration activity to be transparent, including requiring the use of technologies such as the Internet to achieve this goal. Aiming to develop an alternative way to obtain information on the expenditures on works at the Universidade Federal Maranhão (UFMA), this work presents the System of Visualization of Works (SIVIO), a mobile application that uses georeferencing to show the location of these works. The purpose of this tool is to use technologies that facilitate the search and access to information and thus encourage citizens, especially the academic community of UFMA to take more knowledge about the actions of university administration.

Keywords: SIVIO, Government Transparency, Georeferencing, Mobile Application.

Lista de ilustrações

Figura 1 – <i>Pins</i> mostram a localização de um ponto importante no mapa.	17
Figura 2 – Uma estrutura simples em JSON	20
Figura 3 – Etapas de desenvolvimento do SIVIO	23
Figura 4 – Diagrama de Casos de Uso do SIVIO no lado cliente	25
Figura 5 – Diagrama de Casos de Uso do SIVIO no lado servidor	25
Figura 6 – Diagrama de Classe do SIVIO	26
Figura 7 – Tela de pedido de permissão	28
Figura 8 – Código responsável por obter dados do <i>web service</i>	29
Figura 9 – Código responsável por obter dados do <i>web service</i>	30
Figura 10 – Tela Inicial	31
Figura 11 – Após tocar em um <i>pin</i> (a) e tela de detalhes da obra (b).	32
Figura 12 – Uma parte do arquivo JSON utilizado pelo SIVIO	32
Figura 13 – Menu (a) e tela de lista de obras (b).	33

Lista de tabelas

Tabela 1 – Tabela comparativa de ferramentas de acesso à informação	13
---	----

Lista de abreviaturas e siglas

SIVIO	<i>Sistema de Visualização de Obras</i>
UFMA	<i>Universidade Federal do Maranhão</i>
SIG	<i>Sistema de Informação Geográfica</i>
API	<i>Application Programming Interface</i>
SGBD	<i>Sistema Gerenciador de Banco de Dados</i>

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.2	Organização do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Sistema de Informação Geográfica - SIG	15
2.1.1	Georreferenciamento	15
2.1.2	Google Maps	16
2.2	Desenvolvimento Móvel Multi-Plataforma	17
2.2.1	Xamarin	18
2.3	JSON	19
2.4	<i>Threads</i>	20
2.5	Programação Assíncrona	21
3	ESTUDO DE CASO - SIVIO	23
3.1	Requisitos	23
3.2	Modelagem	24
3.2.1	Diagrama de Casos de Uso	24
3.2.2	Diagrama de Classe do SIVIO	24
3.3	Implementação	26
4	CONCLUSÃO	34
4.1	Trabalhos Futuros	34
	REFERÊNCIAS	36

1 Introdução

A Constituição Federal (BRASIL, 1988), no inciso XXXIII do artigo 5º, expressa que "todos têm direito a receber dos órgãos públicos informações de seu interesse particular, ou de interesse coletivo ou geral". Esse inciso foi posteriormente regulado pela lei nº 12.527 (BRASIL, 2011) que "dispõe sobre os procedimentos a serem observados pela União, Estados, Distrito Federal e Municípios, com o fim de garantir o acesso a informações" previstos na Constituição.

A lei nº 12.527, no parágrafo 2º do artigo 8º, vai além e obriga órgãos e entidades públicas a divulgarem "em sítios oficiais da rede mundial de computadores" as "informações de interesse coletivo ou geral por eles produzidas ou custodiadas" (BRASIL, 2011).

As ações tomadas pelos órgãos governamentais após a criação desta lei tornaram mais fáceis o acompanhamento por parte da população com respeito a programas, ações, projetos e obras, fornecendo informações essenciais para os cidadãos sobre as atividades da administração pública brasileira.

Neste aspecto, uma das formas mais acessíveis para consultas de tais informações é pela internet, por meio dos portais de transparência de cada camada do governo, dos órgãos e entes governamentais. O Portal da Transparência do Governo Federal é uma fonte abundante de informações sobre assuntos da gestão pública, possuindo vários filtros, gráficos e opções de pesquisa. Entretanto alguém que acessa casualmente o portal pela primeira vez pode ficar intimidado pela gama de possibilidades apresentadas. Embora não seja um problema em si, o excesso de informações pode ser um impedimento para alguns.

Um outro serviço que fomenta a transparência é o Sistema Eletrônico do Serviço de Informação ao Cidadão (e-SIC) que também funciona na internet e centraliza os pedidos por parte de qualquer pessoa, física ou jurídica ao Poder Executivo Federal, suas entidades vinculadas e empresas estatais. Após feito o cadastro, e o pedido, o órgão tem até 20 dias para responder, prorrogáveis por mais 10 se necessário. É possível recorrer das respostas, ou falta delas.

Além das plataformas *web*, em 14 de novembro de 2016 o Ministério do Planejamento, Desenvolvimento e Gestão, lançou o aplicativo Desenvolve Brasil, que amplia a participação do cidadão na gestão governamental e torna mais transparente as ações do governo em relação ao andamento de obras que recebem recursos do Orçamento Federal. Com a ferramenta é possível verificar informações como situação, localização e investimentos previstos. Integrado às redes sociais, permite o envio de fotos, avaliações e comentários sobre os empreendimentos (PLANEJAMENTO, 2016). O aplicativo está disponível para a plataforma Android (a partir da versão 5.1).

Tabela 1 – Tabela comparativa de ferramentas de acesso à informação

	Pontos fortes	Pontos fracos
Portal da Transparência	- Facilidade de acesso	- Excesso de informação; - Conhecimento prévio sobre a pesquisa
e-SIC	- Liberdade para solicitação de informações (texto corrido) de qualquer tipo	- Espera para obtenção de dados; - Apenas informações sobre o Executivo Federal
SIPAC	- Ampla gama de tipos de processo	- Excesso de informação; - Interface pouco amigável; - Limitada a funcionários e docentes da UFMA
Desenvolve Brasil	- Facilidade de uso - Plataforma móvel; - Uso de geolocalização; - Integração com redes sociais;	- Apenas informações sobre o Executivo Federal; - Não há acesso ao histórico de obras; - Não há informações sobre empresas

Por fim, na esfera estadual, a Universidade Federal do Maranhão (UFMA) também possui o seu próprio portal de transparência denominado Sistema Integrado de Patrimônio, Administração e Contratos, ou SIPAC, onde é possível visualizar todas as informações detalhadas sobre todo tipo de processos administrativos ocorrendo na universidade, entre licitações, ações judiciais, diárias e outros (SINFO, 2017).

A Tabela 1, mostra uma comparação entre as plataformas discutidas anteriormente.

Embora a maioria delas cumpra o papel fundamental de prover informação, ainda há necessidade de se fazer isso de uma forma mais simples e intuitiva. O aplicativo "Desenvolve Brasil" propõe e cumpre a ideia de trazer a transparência para uma plataforma mais interativa, ainda que faça isso com alguns elementos importantes deixados de lado como, por exemplo, quais empresas estão realizando o serviço em determinado empreendimento, e cubra apenas os repasses provenientes do governo federal.

No âmbito da Universidade Federal do Maranhão, em especial ao que se refere às obras, a instituição publica anualmente um relatório do que foi feito no período, bem como de onde recebeu repasses, além de outros assuntos.

Diante dessas situações, introduzimos o Sistema de Visualização de Obras (SIVIO), uma aplicação com o papel de aliar acesso à informação com facilidade e rapidez à visualização dessa informação, no contexto da UFMA.

1.1 Objetivos

Este projeto propõe a criação de uma ferramenta que utilize tecnologias que facilitem o acesso à informação, como a georreferenciamento, para visualização das obras realizadas no âmbito do *campus* do Bacanga (UFMA) e ao mesmo tempo, fornecer, mais detalhadamente, informações a cerca das mesmas para o cidadão. Assim a comunidade acadêmica e outros interessados poderiam ter a seu dispor uma maneira mais simples de saber como os recursos públicos recebidos pela instituição são utilizados, de onde partem,

e por quem são executados, além de poderem acompanhar o andamento da realização dos empreendimentos.

Destacam-se como objetivos específicos deste trabalho:

- Analisar a plataforma de desenvolvimento de aplicativos móveis Xamarin ([PETZOLD, 2016](#));
- Examinar as ferramentas mais utilizadas de acesso a informações públicas;
- Integrar tecnologias de Sistemas de Informações Geográficas ao aplicativo;
- Implementar um aplicativo funcional para auxiliar a população a verificar as obras que estão acontecendo no campus do Bacanga.

1.2 Organização do Trabalho

Para melhor apresentação do conteúdo, este trabalho está organizado em quatro capítulos, conforme os parágrafos a seguir.

O Capítulo 2, Fundamentação Teórica, apresenta o embasamento teórico necessário para o entendimento e desenvolvimento do sistema proposto.

O Capítulo 3, Estudo de Caso, apresenta a modelagem e implementação do sistema proposto.

O Capítulo 4, Conclusão, apresenta as considerações finais acerca do trabalho realizado.

2 Fundamentação Teórica

Este capítulo visa apresentar a fundamentação teórica necessária para a compreensão das técnicas e tecnologias utilizadas no desenvolvimento deste trabalho. Destaca-se entre estas o georreferenciamento, uma importante técnica para localização de pontos ou áreas em um mapa. Também é dada uma visão geral sobre a plataforma Xamarin, utilizada para a criação do aplicativo SIVIO.

2.1 Sistema de Informação Geográfica - SIG

Um Sistema de Informação Geográfica (SIG ou GIS, do inglês *Geographic Information System*), é um sistema computacional capaz de realizar uma grande quantidade de operações sobre informações geográficas. Essas informações geográficas por sua vez se referem a qualquer informação na superfície terrestre (LONGLEY et al., 2005).

Os SIGs podem usar qualquer informação que inclua uma localização, independentemente de como esta é expressa (longitude e latitude, endereço).

Estes sistemas já fazem parte do nosso cotidiano há algum tempo e aplicações que utilizam seus serviços surgem a todo o momento. Como exemplo, há aplicativos que fazem uso da localização atual de uma pessoa para encontrar o caixa eletrônico ou o supermercado mais próximo. Ainda neste aspecto, empresas que prestam serviços de transporte compartilhado tem se popularizado e conseguido se estabelecer, fazendo frente a meios de transporte mais tradicionais como os táxis, por utilizarem SIG para localizar clientes e mostrar o veículo mais próximo a eles. Além disso, muitas pessoas não saem de suas residências sem antes verificar algum aplicativo de monitoramento de tráfego, seja para saber como chegar a determinado lugar ou qual rota está menos congestionado naquele horário. Por fim, mesmo pequenos empresários podem facilmente informar a localização de suas empresas, além de adicionar informações básicas dela, como local e contato, a serviços SIG bem conhecidos.

A seguir veremos uma técnica essencial para os SIGs e que está presente neste trabalho.

2.1.1 Georreferenciamento

Segundo Hill (2009), o escopo de georreferenciamento abrange o significado informal, de se referir à localização utilizando o nome de lugares como se faz em conversas cotidianas, e a representação formal, baseada em coordenadas de latitude e longitude e outros sistemas de referenciamento espacial, que são utilizados em atividades como elaboração de mapas

e navegação. A representação formal são como pegadas geoespaciais - assim chamadas pois nos mostram em um mapa da superfície da Terra, um ponto ou área específico de onde algo está localizado, permitindo que se possa relacionar um lugar com qualquer outro ponto geográfico de interesse.

A maioria do georreferenciamento que encontramos hoje se dá na forma de nome de lugares (HILL, 2009). Quando fazemos uma pesquisa por um nome de algum lugar que queremos ir ou saber mais a respeito, usamos o nome desse lugar, seja uma cidade, ou um endereço mais específico. Entretanto alguns lugares podem ser identificados por vários nomes, ou existem vários lugares com o mesmo nome, fazendo com que o uso de texto para esse tipo de busca não seja o mais eficaz. E o que dizer de áreas localizadas no mar? Para isso é necessário usarmos o georreferenciamento em sua definição formal, ou ainda, usa-la junto com a definição informal para tirar pleno proveito desta.

O SIVIO utiliza o georreferenciamento para mostrar o mapa do *campus* do Bacanga da UFMA, exibindo a localização das instalações e das obras sendo realizadas.

Uma ferramenta de georreferenciamento (além de outras funções ligadas a SIG) bastante difundida, e utilizada no desenvolvimento do SIVIO é o Google Maps.

2.1.2 Google Maps

O Google Maps é um serviço de visualização e pesquisa de mapas na web, desenvolvido pela empresa americana Google e lançado em fevereiro de 2003. Ele oferece imagens de satélites, mapas de ruas, visão panorâmica de 360° de ruas, verificação das condições de tráfego em tempo real ou em um determinado horário, e planejamento de rota para viagens a pé, de carro, de bicicleta e por transporte público. (GOOGLE, 2017)

Além disso, desde junho de 2005, está disponível a Interface de Programação de Aplicação (API, do inglês *Application Programming Interface*) do Google Maps que permite que desenvolvedores integrem esse serviço em seus *websites*. Hoje em dia também está disponível para uso em aplicações móveis. Assim é possível construir outras aplicações que possam tirar proveito de uma base de mapas já consolidada e com possibilidades de customização. Esta API é gratuita para uso comercial, desde que o site ou aplicativo possa ser acessado gratuitamente. A cota diária nesse caso é de 25 mil carregamentos de mapas. (GOOGLE, 2017)

Um elemento importante do Google Maps utilizado para caracterizar um ponto de interesse no mapa é o *pin*. Ao clicar nele informações sobre a distância deste lugar para um outro ponto, entre outros detalhes são mostrados. Um Pin pode ter várias cores e ícones indicando que certo ponto pertence a um grupo. Na Figura 1 são mostrados vários tipos de *pins*: verde se refere a instalações com fins recreativos ou esportivos, como ginásios, praças e parques; cinza é atribuído a prédios de órgãos públicos; o *pin* maior em vermelho

é um ponto clicado pelo usuário. Este componente é a forma como o SIVIO mostra o local das obras correntes no *campus* do Bacanga.

Figura 1 – *Pins* mostram a localização de um ponto importante no mapa.



Fonte: O autor (2018)

2.2 Desenvolvimento Móvel Multi-Plataforma

Embora ainda bastante utilizado para tarefas que necessitam de um teclado e uma tela grande, o computador de mesa tem dado cada vez mais espaço a dispositivos menores quando se fala de computação pessoal, particularmente para informações rápidas, uso de mídias e redes sociais. *Tablets* e *smartphones* possuem um paradigma de interação com usuário fundamentalmente diferente daquele encontrados em computadores de mesa, baseado primariamente no toque, com um teclado que aparece apenas quando necessário (PETZOLD, 2016).

Petzold (2016) destaca ainda, que embora o mercado de telefonia móvel tenha um potencial para mudanças rápidas, atualmente duas grandes plataformas de telefone móvel e *tablet* dominam:

- A família Apple de iPhones e iPads, todos os quais executam o sistema operacional iOS.
- O sistema operacional Android, desenvolvido pelo Google com base no *kernel* do Linux, que é executado em vários telefones móveis e tablets.

Há também uma terceira plataforma de desenvolvimento móvel, que não é tão popular quanto o iOS e o Android, mas envolve uma empresa com um histórico forte na indústria de computadores pessoais (PETZOLD, 2016):

- Windows Phone da Microsoft e Windows 10 *Mobile*.

Segundo Petzold(2016), a estratégia ótima para os desenvolvedores seria almejar mais do que uma dessas plataformas, mas isso não é fácil devido a quatro problemas:

- Diferentes paradigmas de interface de usuário, no que se refere à navegação e apresentação de dados e menus, e até mesmo abordagem ao toque da tela.
- Diferentes ambientes de desenvolvimento.
- Diferentes interfaces de programação.
- Diferentes linguagens de programação.

Muitas soluções para programação multi-plataforma surgiram nos últimos anos, sendo a plataforma Xamarin uma das mais populares, a qual será utilizada neste trabalho.

2.2.1 Xamarin

Muitas das soluções existentes atualmente para desenvolvimento entre plataformas se utilizam de tecnologias *Web* (tais como HTML, CSS e Javascript) para transformar aplicativos Web em aplicativos para as plataformas mais populares entre dispositivos móveis (os sistemas operacionais Android e iOS). O problema deste tipo de abordagem é que o desempenho deixa a desejar e a experiência de usuário bem como a interface ficam mais próximo de uma página *Web* do que de um aplicativo nativo (ALTEXSOFT, 2018).

Em março de 2016, a Microsoft adquiriu a empresa Xamarin (que também dá nome ao seu principal produto: a plataforma Xamarin), com a qual possuía parceria anteriormente. Entretanto para utilizar esses serviços era necessário pagar. A aquisição derrubou a barreira financeira e permitiu com que a Microsoft pudesse promover a abordagem de compartilhamento de código para desenvolvimento entre plataformas.

A grande diferença entre Xamarin e as abordagens baseadas em tecnologias web é que Xamarin proporciona o desenvolvimento de aplicações de maneira nativa para os sistemas operacionais iOS e Android, além do Windows Phone, utilizando a linguagem C# (PETZOLD, 2016). Isso significa que os aplicativos criados com o Xamarin contêm controles de interface do usuário nativos padrão. Os aplicativos não têm apenas a aparência que o usuário final espera, mas também se comportam dessa forma. (MICROSOFT, 2018)

Uma grande vantagem de utilizar uma única linguagem de programação vem da capacidade de compartilhar código entre os aplicativos (PETZOLD, 2016). Ainda assim, para aplicações que possuem necessidades de escrever código específico para cada plataforma, mas que ainda tenham uma quantidade grande de código compartilhável (como as regras de negócio), o Xamarin fornece a possibilidade de escrever código individual.

No SIVIO foram necessários pouco mais de 100 linhas de código específico para a plataforma Android, a maior parte requerida para tratamento de permissões que precisam ser manipuladas diretamente por cada sistema operacional. No caso do SIVIO, é necessário ter acesso à localização do usuário para utilizar o Google Maps.

Um ponto significativo em relação ao funcionamento de grande parte dos aplicativos, em especial aqueles que necessitam de acesso à *internet* para buscar ou enviar dados, é que estes precisam de um modo para fazer isso de tal forma que tanto o aplicativo como o servidor com o qual se comunica consigam interpretar a informação enviada ou recebida, independente de qual foi a linguagem de programação usada na construção de ambos os lados.

Para isso, utilizam-se formatos de trocas de dados que sejam leves e simples como a Extensible Markup Language (XML) e O *Javascript Object Notation* (JSON), este utilizado na comunicação de dados do aplicativo SIVIO com o lado servidor, de onde obtém os seus dados.

2.3 JSON

O JSON é um formato de arquivo independente de linguagem. Embora tenha sua origem derivada no Javascript, ele utiliza muitas convenções que são familiares a outras linguagens (JSON, 2018).

O JSON é constituído de duas estruturas:

- Uma coleção de pares nome/valor, que pode ser caracterizado como um objeto, estrutura ou dicionário em algumas linguagens.
- Uma lista ordenada de valores, que na maioria das linguagens é caracterizado como um *array*, vetor, lista ou sequência.

Praticamente todas as linguagens de programação suportam essas estruturas de alguma forma, fazendo com que ele seja um formato ideal para a troca de dados. JSON utiliza as nomenclaturas objeto para coleção de pares/valor e *array* para lista ordenada. Um objeto começa com uma chave de abertura (`{`) e termina com uma chave de fechamento (`}`). Dentro delas cada nome (entre aspas) é seguido por dois pontos (`:`) e um valor associado ao nome. Os valores podem ser:

- Uma cadeia de caracteres (*string*);
- Um número;
- Um objeto;
- Um *array*;
- Um valor booleano (*true* ou *false*);
- *null*

Os pares nome/valor são separados por vírgula. Uma *array* é uma coleção de valores ordenados. O *array* inicia com um colchete de abertura ([]) e termina com um colchete de fechamento (]). Os valores são separados por vírgula (JSON, 2018). A Figura 2 exemplifica essas definições.

Figura 2 – Uma estrutura simples em JSON

```
{
  "professor": {
    "nome": "Fulano",
    "idade": 34,
    "disciplinas": [
      "Estrutura de Dados",
      "Engenharia de Software"
    ]
  }
}
```

Fonte: O autor (2018)

2.4 *Threads*

O conceito de *threads* é intrínseco à ideia de processos. Tanenbaum (2010) define processo como uma abstração de um programa em execução. Eles mantêm a capacidade de operações (pseudo)concorrentes, mesmo quando há apenas uma CPU disponível. Eles transformam uma única CPU em diversas CPUs virtuais.

Em sistemas operacionais tradicionais, cada processo tem um espaço de endereçamento e um único *thread* de controle. No entanto, frequentemente há situações em que é desejável ter múltiplos *threads* de controle no mesmo espaço de endereçamento

executando em quase-paralelo, como se eles fossem processos separados (TANENBAUM, 2010)).

A principal razão para o uso de *threads*, ou mini-processos, é que em muitas aplicações ocorrem múltiplas atividades ao mesmo tempo. O modelo de programação se torna mais simples se decomposmos uma aplicação em múltiplos *threads* sequenciais que executam em quase-paralelo (TANENBAUM, 2010).

O conceito de *thread* é melhor compreendido utilizando um exemplo onde seu uso é crucial: a programação assíncrona.

2.5 Programação Assíncrona

Uma de nossas principais referências para avaliarmos um programa que baixamos se trata do seu desempenho. Um aplicativo que apresenta travamentos, demora para concluir uma ação e fechamentos repentinos pode ser rapidamente deletado e mal avaliado. Para isso, o desenvolvedor precisa estar alerta à experiência do usuário. Ao clicar em um botão, o usuário precisa ter a resposta imediata daquela ação, ou se a ação necessita de uma espera maior é importante mostrar que o aplicativo está buscando a informação. Isso pode ser feito, por exemplo, colocando uma contagem em forma de porcentagem, tudo para que o usuário não ache que o programa parou de funcionar. Para isso utiliza-se técnicas como a programação assíncrona para contornar problemas de desempenho ou de resposta por parte do programa.

As interfaces gráficas têm uma pequena peculiaridade com uma importante consequência: A entrada do usuário em um aplicativo deve ser processada sequencialmente. Independentemente de os eventos de entrada do usuário virem de um teclado, mouse ou toque, cada evento deve ser completamente processado por um aplicativo antes que se obtenha do sistema operacional o próximo evento de entrada de usuário (PETZOLD, 2016).

As consequências dessa restrição são sérias. Toda entrada de usuário para uma aplicação deve ser processada em um simples *thread* de execução. Todo código conectado com a interface de usuário da aplicação é restrita a um único *thread*, conhecido como *thread* principal ou *thread* da interface de usuário (PETZOLD, 2016).

Petzold (2016) explica ainda que, por esta razão quaisquer operações prolongadas que uma aplicação deva executar, devem ser relegadas a *threads* secundários, os quais trabalham “em segundo plano” e não interferem na capacidade de resposta do *thread* principal.

Em anos recentes o processamento assíncrono tem se tornado mais onipresente. Sem dúvida, o futuro da computação irá envolver ainda mais a computação assíncrona e o

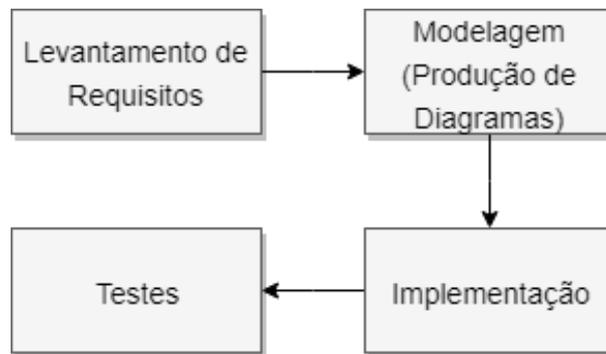
processamento paralelo, particularmente com o aumento dos processadores com múltiplos núcleos (PETZOLD, 2016). O Xamarin, a plataforma utilizada na criação do SIVIO, faz uso desta técnica.

Este capítulo apresentou a fundamentação teórica empregada na implementação do SIVIO. O próximo capítulo explanará como as técnicas e tecnologias consideradas foram utilizadas no desenvolvimento da aplicação e mostrará como elas cumprem o propósito principal de fornecer um serviço de utilidade pública que visa a transparência dos gastos da administração da UFMA com as obras realizadas no *campus* do Bacanga.

3 Estudo de Caso - SIVIO

Ao se planejar a criação de um aplicativo, é essencial expor e escrever o que se espera dele e quais serão as suas principais funcionalidades, especificando seus requisitos. A partir dessa etapa é possível criar algumas representações visuais de elementos que auxiliam a implementação de um sistema, como o diagrama de casos de uso e o diagrama de classe. Por fim, utilizando os requisitos e os diagramas pode-se iniciar a etapa de implementação. As etapas descritas acima foram utilizadas no desenvolvimento do SIVIO e são ilustradas na Figura 3.

Figura 3 – Etapas de desenvolvimento do SIVIO



Fonte: O Autor (2018)

3.1 Requisitos

A seguir é descrito, em linguagem natural, os principais requisitos em nível de usuário do SIVIO, que será composto por uma aplicação móvel, e uma aplicação *web*.

A página inicial da aplicação móvel consistirá de um mapa e um menu. A aplicação iniciará mostrando o mapa do *campus* do Bacanga da UFMA. O mapa mostrará as instalações (prédios, quadras etc) do *campus*. Essas instalações serão destacadas e identificadas por nome. No mapa também serão mostrados as obras ativas no *campus*, identificadas por meio de *pins*. Ao tocar sobre um dos *pins* destacadas no mapa, uma caixa de diálogo será exibida com informações resumidas sobre a obra em questão: a instalação a qual ela está vinculada e o tipo de obra sendo realizada (construção, reforma, recuperação ou ampliação). Ao tocar sobre a caixa de diálogo, outra tela será exibida, desta vez apresentando os detalhes sobre a obra. Esses detalhes incluem sua descrição, a

data de início, o prazo para conclusão, o andamento (em porcentagem), o custo, a empresa contratada (em caso de execução indireta) e a origem dos recursos.

Além de poder acessar as obras ativas diretamente pela tela inicial, o usuário poderá obter essas informações por meio de uma lista contendo as obras ativas e, também, as que já foram concluídas, por meio do menu. Esta lista será exibida em ordem decrescente (considerando a data de início da obra). Assim como os *pins*, os itens da lista terão uma breve descrição sobre cada obra, bem como o seu tipo e a data de início. Por fim, ao clicar sobre um item da lista, uma tela mostrando os detalhes sobre aquela obra específica será exibida.

Ainda no menu será possível obter a lista das empresas que realizaram as obras.

Por meio de uma aplicação *web*, o administrador gerenciará o cadastro das obra com suas respectivas informações, incluindo, modificando e excluindo itens. Após confirmado o cadastro desses itens, essas informações estarão disponíveis para o aplicativo por meio de um *web service*.

A próxima seção mostra a modelagem do sistema, baseada nos requisitos definidos anteriormente.

3.2 Modelagem

A seguir, serão apresentados dois importantes componentes da modelagem de *software*: o diagrama de casos de uso e o diagrama de classe.

3.2.1 Diagrama de Casos de Uso

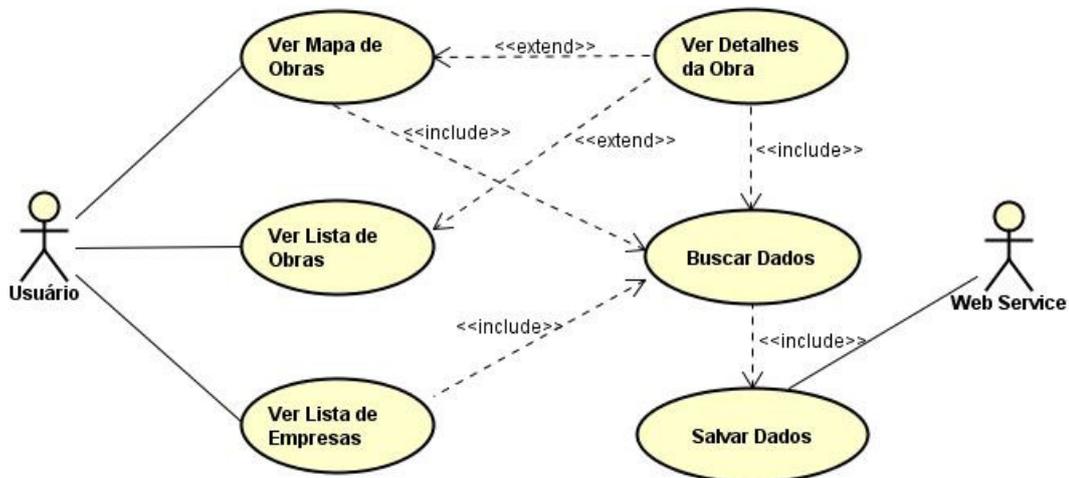
A Figura 4 mostra o diagrama de casos de uso do SIVIO no lado cliente (aplicação móvel) e a Figura 5 mostra o diagrama de casos de uso lado servidor (aplicação *web*), descrevendo os comportamentos do sistema em relação aos atores que interagem com cada componente.

Os diagramas de caso de uso tanto no lado cliente quanto no lado servidor possuem interação com o ator *Web Service*. Este ator é um componente que permite com que aplicação *Web* possa enviar dados para outra plataforma, representada neste caso pela aplicação cliente.

3.2.2 Diagrama de Classe do SIVIO

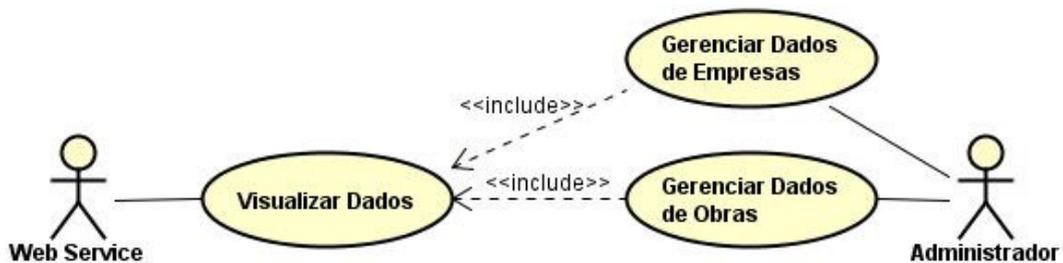
Na modelagem do SIVIO foi utilizado o padrão de projeto *Model-View-ViewModel* (MVVM) cujo objetivo é separar as responsabilidades de uma aplicação. A *View* é a camada de apresentação, ou interface com o usuário. A camada *Model* é responsável pela lógica de

Figura 4 – Diagrama de Casos de Uso do SIVIO no lado cliente



Fonte: O Autor (2018)

Figura 5 – Diagrama de Casos de Uso do SIVIO no lado servidor



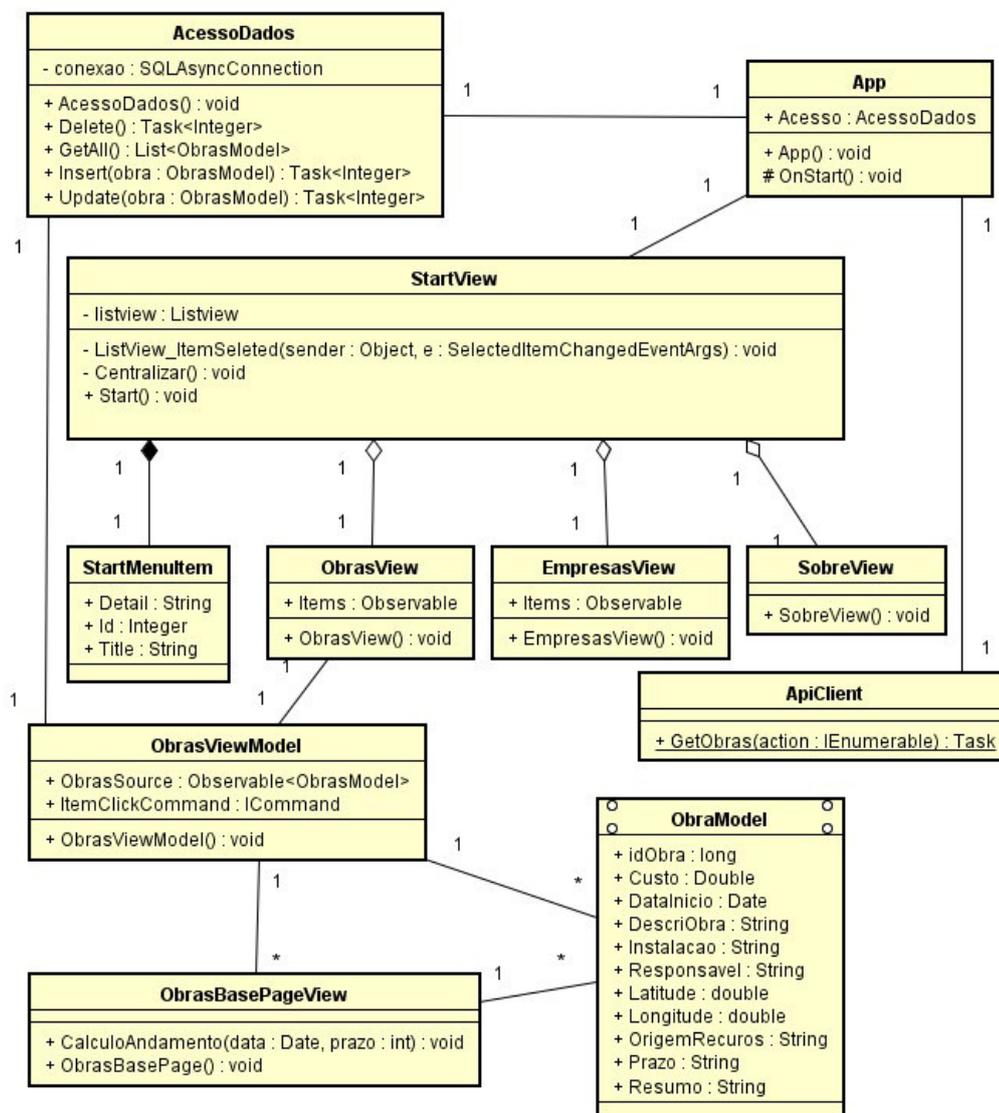
Fonte: O Autor (2018)

negócio. Por sua vez, a *ViewModel* é a camada intermediária, e trata da comunicação entre *View* e *Model*, também se preocupando com a lógica de apresentação que será exposta na *View* (PETZOLD, 2016).

A Figura 6 mostra o diagrama de classe seguindo o padrão MVVM.

A próxima seção mostrará como os requisitos, com auxílio dos diagramas, foram implementados.

Figura 6 – Diagrama de Classe do SIVIO



Fonte: O Autor (2018)

3.3 Implementação

O SIVIO foi implementado utilizando a plataforma Xamarin por meio do ambiente de desenvolvimento integrado (IDE) Visual Studio da Microsoft. Embora esta tecnologia permita gerar aplicativos nativos para as três grandes plataformas móveis existentes atualmente (iOS, Android e Windows Phone), para fins da apresentação das funcionalidades do SIVIO, neste trabalho será utilizado apenas o aplicativo gerado para a plataforma Android.

Um dos pontos de projeto centrais da arquitetura de segurança do Android é que, por padrão, nenhum aplicativo tem permissão de realizar nenhuma operação que

prejudique outros aplicativos, o sistema operacional ou o usuário. Isso abrange a leitura e a gravação de dados privados do usuário (como contatos ou e-mails), leitura ou gravação dos arquivos de outro aplicativo, realização de acesso de rede, manter o dispositivo ativo etc (ANDROID, 2018a).

No Android, até a versão 5.1 (lançada em 7 de julho de 2015) , antes de baixar um aplicativo na Play Store (loja virtual do Google para obtenção e compra de aplicativos) o usuário tinha uma lista completa de todas as permissões que o aplicativo iria necessitar.

A partir da versão 6.0 do Android (lançada em 5 de outubro de 2015), as permissões foram simplificadas, e passou-se a utilizar grupos de permissões (passaram de 24 para apenas 8). Também foi introduzida uma nova forma de pedir por essas permissões para o usuário, onde essa solicitação ocorre em tempo de execução, ou seja, ao abrir um aplicativo para utilizá-lo (ANDROID, 2018a).

Para os testes, o SIVIO foi executado tanto na versão 5.1 quanto na 6.0 do Android, sendo que as imagens encontradas neste trabalho se referem à versão 6.0. Essas versões foram escolhidas pois são as variantes mais utilizadas em aparelhos no mundo (ANDROID, 2018b) e também por poder demonstrar como o aplicativo lida com diferentes abordagens de cessão de permissões. O SIVIO pode ser executado em qualquer aparelho que execute o Android a partir da versão 5.1, sendo a 8.1 a mais recente (lançada em 5 de dezembro de 2017).

No SIVIO foi utilizado o grupo de permissões "Localização", necessário para uso da API do Google Maps, embora a aplicação em si não faça uso da localização do usuário. A Figura 7 mostra a tela na qual a permissão é solicitada.

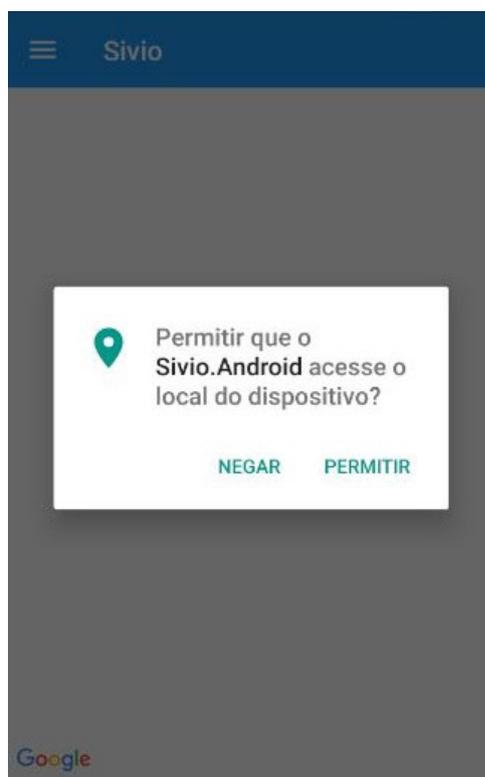
Após a cessão da permissão, o SIVIO busca no *web service* as informações sobre as obras. A Figura 8 mostra o método responsável por acessar o *web service*, tratar os dados e disponibilizá-los para o aplicativo que posteriormente irá atualizar seu banco de dados local chamando este método.

Em seguida, será carregada a tela inicial com o mapa, o qual conterá os *pins* - que exibem os locais das obras ativas (e também daquelas que ainda irão iniciar) no *campus* do Bacanga. Cada um contem as informações que foram buscadas no banco de dados. A Figura 9 mostra o código correspondente a esta ação.

A tela inicial com o mapa e os *pins* é apresentada na Figura 10. Nela também pode-se observar que uma mensagem é mostrada para auxiliar o usuário a como obter detalhes sobre as obras. Essa mensagem é mostrada toda vez que o aplicativo é aberto e dura cerca de 4 segundos.

Quando um usuário toca em um desses *pins*, uma caixa de diálogo aparece com duas informações sobre a obra: a instalação e o tipo de obra, como mostra a Figura 11a.

Figura 7 – Tela de pedido de permissão



Fonte: O Autor (2018)

O usuário pode pressionar qualquer parte da tela para que a caixa de diálogo desapareça, e continuar navegando pelo mapa.

Caso pressione na caixa de diálogo, ele será redirecionado para outra tela que mostrará os pormenores da obra selecionada. As informações detalhadas são mostradas na Figura 11b. A partir desta tela é possível voltar para a tela inicial por meio da seta presente no canto superior esquerdo.

Todas as informações são obtidas a partir do servidor, com exceção do andamento que é calculado no âmbito do aplicativo, comparando a data atual com a data de início da obra e também com o prazo. Com essas informações o aplicativo define em quais condições a obra se encontra, podendo então expor se a obra já iniciou ou terminou, ou ainda o andamento de uma obra iniciada (expresso em porcentagem).

A Figura 12 exemplifica uma parte da estrutura do arquivo JSON que é enviado do servidor para o aplicativo após a solicitação deste. Além das informações das obras, esses dados também expressam a localização dos *pins* (georreferenciamento por meio da latitude e longitude).

Figura 8 – Código responsável por obter dados do *web service*

```
public static async Task GetObras(Action<IEnumerable<ObrasModel>> action)
{
    //cria nova conexão HTTP
    HttpClient httpClient = new HttpClient();
    try
    {
        //tenta acessar o web service
        HttpResponseMessage json = await httpClient.GetAsync("https://api.myjson.com/bins/bkfis");
        //caso obtenha sucesso a string obtida é automaticamente deserializada para corresponder
        //com o modelo de dados da classe obra
        if (json.StatusCode == System.Net.HttpStatusCode.OK)
        {
            IEnumerable<Models.ObrasModel> obras = JsonConvert
                .DeserializeObject<IEnumerable<ObrasModel>>(await json.Content.ReadAsStringAsync());
            action(obras);
        }
    }
    catch (HttpRequestException e)
    {
        DependencyService.Get<IMessage>().Alert(e.InnerException.Message);
    }

    httpClient.Dispose();
}
```

Fonte: O Autor (2018)

Uma outra forma de visualizar quais obras estão sendo realizadas é por meio do menu (Figura 13a). Além disso também será possível visualizar a lista de empresas que prestaram serviços na UFMA e informações sobre o aplicativo.

Ao selecionar a opção "Obras" o usuário será redirecionado para uma página com uma lista de obras ativas (representadas pelos *pins* na página inicial) e também das que já terminaram. Além disso sempre que o usuário acessar esta página pela primeira vez, será mostrado uma mensagem informando que ele pode tocar em cada item para saber mais detalhes, conforme a Figura 13b. Essa ação tem o mesmo efeito de clicar na caixa de diálogo de um *pin* da tela inicial, levando o usuário para a página de detalhes da obra como mostrado na Figura 11b.

Todas as telas acessadas a partir da página inicial e do menu possuem no canto superior esquerdo, uma seta de retorno usada para voltar à página anteriormente visitada.

Este capítulo expôs as etapas necessárias para o desenvolvimento do SIVIO, desde a definição de suas funcionalidades, bem como a modelagem do sistema e, por fim, o produto resultante.

O capítulo seguinte, conclui este trabalho e avalia se os objetivos propostos foram cumpridos e propõe trabalhos futuros.

Figura 9 – Código responsável por obter dados do *web service*

```
Task.Run(async () => {
    //Itera sobre os itens (obras) no banco de dados local
    foreach (Models.ObrasModel item in await App.Acesso.GetAll())
    {
        //cria um novo pin para cada item
        var pin = new Pin
        {
            Type = PinType.Place,
            Position = new Position(item.Latitude, item.Longitude),
            Label = item.Instalacao + ": " + item.ResumoObra,
            Address = "Toque para ver Detalhes",
        };

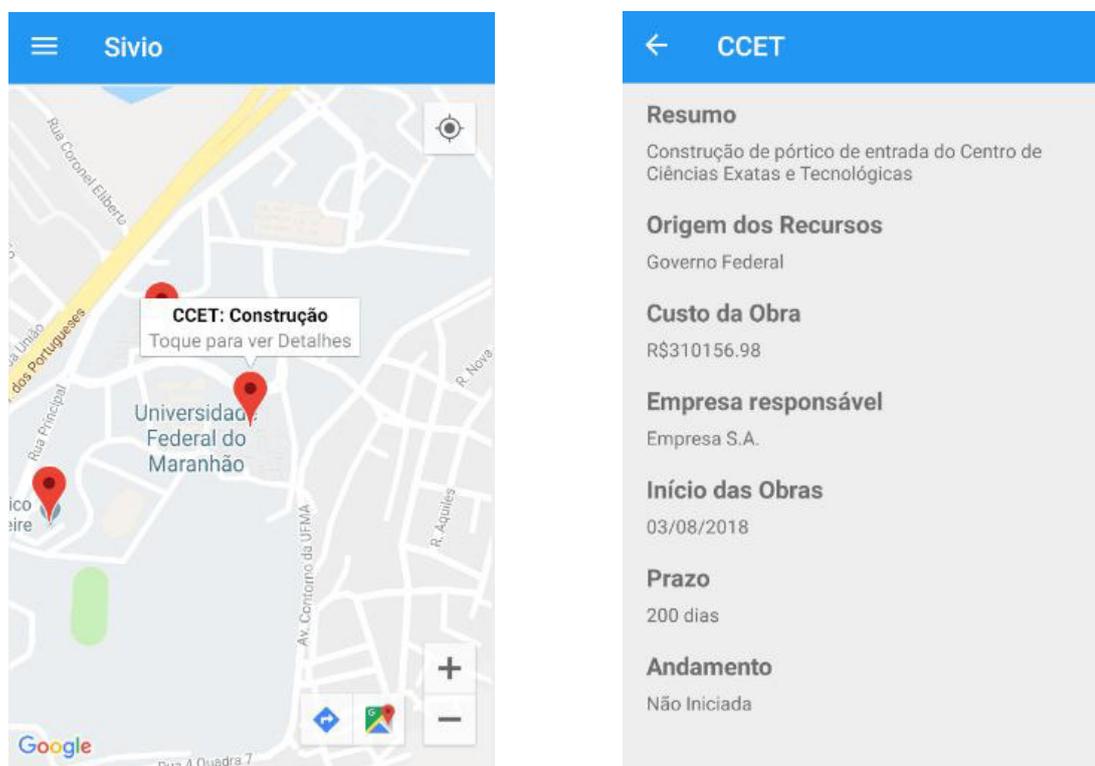
        //adiciona uma ação ao pin responsável por abrir a página
        //detalhes
        pin.Clicked += (sender, e) =>
        {
            var showItem = item;
            var page = new ObrasBasePage(showItem);
            App.MasterDetail.Detail.Navigation.PushAsync(page);
        };
        ufmaMap.Pins.Add(pin);
    }
    Device.BeginInvokeOnMainThread(() =>
    {
        //Exibe mensagem de auxílio ao usuário
        DependencyService.Get<IMessage>().
            Alert("Clique nos Pins para ver Detalhes");
    });
});
```

Fonte: O Autor (2018)

Figura 10 – Tela Inicial



Fonte: O Autor (2018)

Figura 11 – Após tocar em um *pin* (a) e tela de detalhes da obra (b).

(a)

(b)

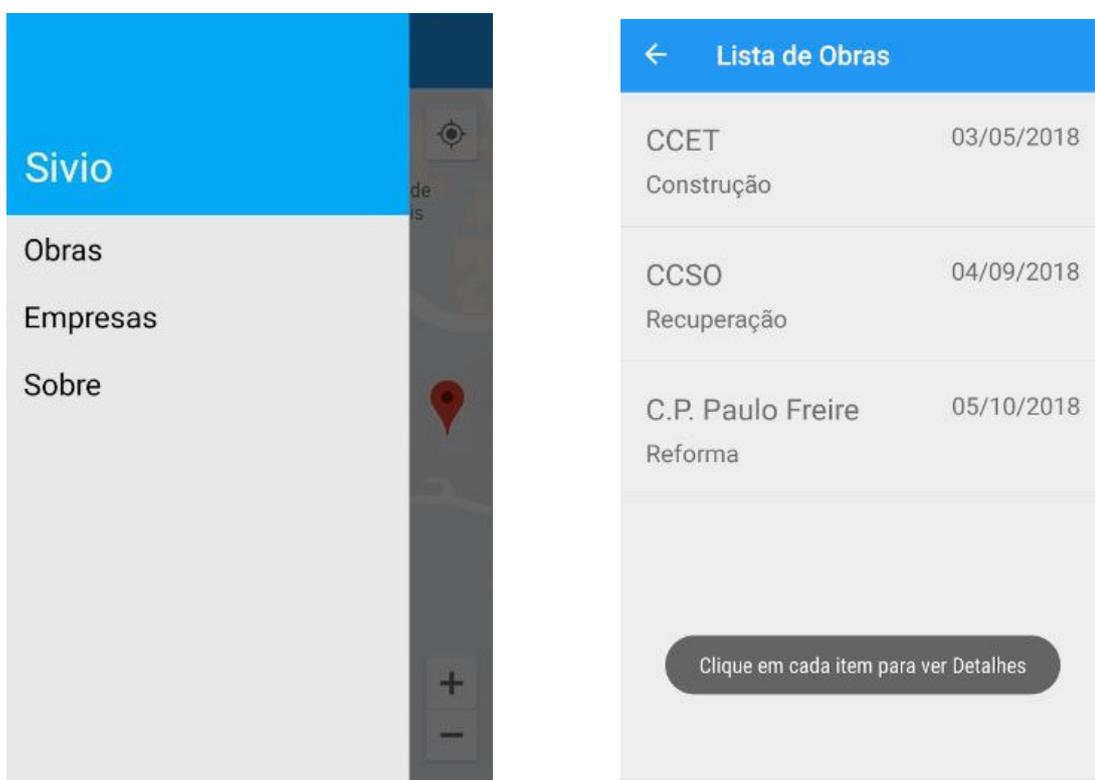
Fonte: O Autor (2018)

Figura 12 – Uma parte do arquivo JSON utilizado pelo SIVIO

```
{
  "IdObra": 5,
  "Instalacao": "CCSO",
  "ResumoObra": "Recuperação",
  "DescriObra": "Pintura da faixa exterior do Centro de Ciências Sociais",
  "Responsavel": "Empresa S.A.",
  "Custo": 160590.78,
  "OrigemRecursos": "Governo Federal",
  "DataInicio": "09/04/2018",
  "Prazo": 180,
  "Latitude": -2.556582,
  "Longitude": -44.30968
},
```

Fonte: O Autor (2018)

Figura 13 – Menu (a) e tela de lista de obras (b).



(a)

(b)

Fonte: O Autor (2018)

4 Conclusão

Essa monografia se propôs a desenvolver uma ferramenta que pudesse aumentar a transparência com respeito aos gastos relacionados às obras realizadas no *campus* do Bacanga, aliando a obtenção de informações sobre essas obras, com facilidade de acessá-las. Para tanto, apresentamos no Capítulo 2 algumas tecnologias que favorecem a apresentação de informações de uma maneira mais interativa, como a utilização de mapas, georreferenciamento e o paradigma de desenvolvimento móvel. No Capítulo 3 pudemos ver como essas tecnologias foram combinadas para desenvolver a solução proposta. O SIVIO pôde cumprir seu objetivo de simplificar o acesso à informação por utilizar um mapa e não necessitar que o usuário insira informações pelo teclado, só tendo o trabalho de tocar a tela e ler aquilo que lhe interessa.

Porém, o sistema ainda possui algumas limitações. A próxima seção propõe melhorias e funcionalidades adicionais.

4.1 Trabalhos Futuros

A visualização de obras que dá nome ao sistema foi o principal componente da etapa de implementação, e foi possível demonstrar grande parte de seu potencial. Entretanto, o lado servidor não foi implementado. Em seu lugar, para fins de demonstração, foi utilizado um *website* que simula um *web service* para consumo da aplicação, mas com o conteúdo JSON gerado manualmente.

Além disso, o aplicativo poderá ser aperfeiçoado, implementado a técnica de notificações *push*, utilizada para avisar um usuário de alguma atualização ocorrida em um aplicativo. No caso do SIVIO, o usuário seria avisado quando uma nova obra fosse cadastrada.

Também é possível automatizar a obtenção de informações através da técnica de *Web scrapping*, que permite que se monitore uma fonte de dados e se extraia dela informações que podem ser então alimentada no banco de dados do servidor, para posteriormente serem consumidas pelo aplicativo.

Por fim, a opção de visualização das empresas poderia ser enriquecida, fornecendo mais detalhes sobre cada empresa, como quantas e quais obras ela já realizou e quanto já recebeu como contrapartida pelos empreendimentos.

Portanto, não se pode encerrar por aqui uma questão tão relevante e que poderá ainda contribuir bastante para o aprimoramento de aspectos sociais como a transparência dos gastos da administração pública, não só no âmbito do *campus* do Bacanga, mas que

também poderia ser estendida para outros *campi* da UFMA, ou ainda adaptada para uso em outras instituições, órgãos e entidades governamentais.

Referências

- ALTEXSOFT. *Xamarin vs React Native vs Ionic: Cross-platform Mobile Frameworks Comparison*. 2018. <<https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-cross-platform-mobile-frameworks-comparison/>>. Acesso em 26 de Junho de 2018. Citado na página 18.
- ANDROID. *Permissões do sistema*. 2018. <<https://developer.android.com/guide/topics/security/permissions?hl=pt-br>>. Acesso em 26 de Junho de 2018. Citado na página 27.
- ANDROID. *Versões da plataforma*. 2018. <<https://developer.android.com/about/dashboards/?hl=pt-br>>. Acesso em 27 de Junho de 2018. Citado na página 27.
- BRASIL. *Constituição da República Federativa do Brasil de 1988*. 1988. <http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>. Acesso em 25 de Outubro de 2017. Citado na página 12.
- BRASIL. *Lei nº 12.527, de 18 de Novembro de 2011. Regula o acesso a informações previsto no inciso XXXIII do art. 5º, no inciso II do § 3º do art. 37 e no § 2º do art. 216 da Constituição Federal; altera a Lei no 8.112, de 11 de dezembro de 1990; revoga a Lei no 11.111, de 5 de maio de 2005, e dispositivos da Lei no 8.159, de 8 de janeiro de 1991; e dá outras providências*. 2011. <http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2011/Lei/L12527.htm>. Acesso em 25 de Outubro de 2017. Citado na página 12.
- GOOGLE. *Google Maps API - FAQ*. 2017. <https://developers.google.com/maps/faq?csw=1#top_of_page>. Acesso em 19 de Junho de 2018. Citado na página 16.
- HILL, L. L. *Georeferencing: The geographic associations of information*. [S.l.]: Mit Press, 2009. Citado 2 vezes nas páginas 15 e 16.
- JSON. *Introdução ao JSON*. 2018. <<https://www.json.org/json-pt.html>>. Acesso em 17 de Junho de 2018. Citado 2 vezes nas páginas 19 e 20.
- LONGLEY, P. A.; GOODCHILD, M. F.; MAGUIRE, D. J.; RHIND, D. W. *Geographic information systems and science*. [S.l.]: John Wiley & Sons, 2005. Citado na página 15.
- MICROSOFT. *Ferramentas do Visual Studio para Xamarin*. 2018. <<https://visualstudio.microsoft.com/pt-br/xamarin/>>. Acesso em 26 de Junho de 2018. Citado na página 18.
- PETZOLD, C. *Creating Mobile Apps with Xamarin. Forms Edition 3*. [S.l.]: Microsoft Press, 2016. Citado 7 vezes nas páginas 14, 17, 18, 19, 21, 22 e 25.
- PLANEJAMENTO. *Planejamento lança Aplicativo Desenvolve Brasil*. 2016. <<http://www.planejamento.gov.br/noticias/planejamento-lanca-aplicativo-desenvolve-brasil>>. Acesso em 30 de Outubro de 2017. Citado na página 12.
- SINFO. *SIPAC - Sistema Integrado de Patrimônio, Administração e Contratos*. 2017. <https://docs.info.ufrn.br/doku.php?id=suporte:sipac:visao_geral>. Acesso em 1 de Novembro de 2017. Citado na página 13.

TANENBAUM, A. S. *Sistemas Operacionais Modernos 3a. Edição*. [S.l.]: Editora Prentice-Hall, 2010. Citado 2 vezes nas páginas [20](#) e [21](#).