

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LISANDRO PROCEDI

Avaliação do framework Xamarin.Forms para desenvolvimento de aplicativos móveis multiplataforma, criando uma aplicação real

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Leandro Krug Wives

Porto Alegre
Novembro de 2016

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitor: Prof. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Sérgio Luis Cechin

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Gostaria de agradecer toda minha família, especialmente meus pais Natal e Anair por todo o apoio e compreensão recebidos durante estes longos anos de estudos. Também agradecer meus irmãos Evandro e Adriana que sempre foram grandes exemplos de dedicação e superação. Agradecer meus tios, Leco e Délia, pela acolhida aqui em Porto Alegre e pelas longas rodas de mate nos finais de tarde. E por fim um agradecimento especial à Naihana, que me acompanha a bastante tempo sempre dando o melhor de si para eu conseguir alcançar mais este objetivo.

Quero agradecer também a todos os meus professores, em especial aos professores do Instituto de Informática da UFRGS por todo o conhecimento compartilhado e os bons momentos vivenciados durante a graduação.

Um agradecimento especial aos meus colegas de curso, pelo apoio nas jornadas de estudo e pela parceria em todos os momentos que passamos juntos, com certeza vocês fazem parte dessa conquista.

RESUMO

O propósito geral deste trabalho é explorar o framework Xamarin.Forms a fim de verificar a eficiência do modelo de desenvolvimento onde a escrita do código é feita em uma única linguagem, e o aplicativo resultado roda em diferentes plataformas móveis. Além disso, avalia-se como o Xamarin.Forms torna transparente os detalhes de implementação de cada plataforma bem como se existem áreas onde o desenvolvedor deve conhecer mais a fundo a plataforma onde o aplicativo final vai rodar. Para tanto, este trabalho descreve o desenvolvimento de um aplicativo para a interpretação de laudos de análise de solo com um único código base, mas que executa nas principais plataformas móveis existentes no mercado.

Palavras-chave: Xamarin.Forms. Plataformas móveis. Aplicativo. Análise de Solo.

Xamarin.Forms framework for cross-platform mobile application development, creating a real application

ABSTRACT

The general purpose of this work is to explore the Xamarin.Forms framework in order to verify the efficiency of the development model where writing the code is done in a single language, and the result application runs on different mobile platforms. In addition, it evaluates how Xamarin.Forms makes transparent the implementation details of each platform as well whether there are areas where the developer should know more or not about the platform where the application will run. In this sense, this work presents the development of an application for interpretation of soil analysis reports with a single base code, but that can run on the main mobile platforms existing in the marketplace.

Keywords: Xamarin.Forms. Mobile platforms. App. Soil Analysis.

LISTA DE FIGURAS

Figura 1: Esquema da plataforma Xamarin.....	3
Figura 2: Xamarin + Xamarin.Forms	4
Figura 3: Tipos de Pages mais utilizadas.....	6
Figura 4: Tipo de Layouts mais utilizados	6
Figura 5: Esquemático da arquitetura MVVM	7
Figura 6: Exemplo de utilização da propriedade BindingContext	8
Figura 7: Tela do TFS com as atividades de uma iteração	13
Figura 8: Protótipos de interface de usuário	15
Figura 9: Modelo ER da base de dados	16
Figura 10: Representação da estrutura básica do projeto	17
Figura 11: Divisão da estrutura do projeto conforme modelo MVVM.....	18
Figura 12: Laudo de análise de solo	19
Figura 13: Tela inicial do aplicativo.....	21
Figura 14: Tela do cadastro de culturas.....	22
Figura 15: Tela do cadastro de culturas preenchida	23
Figura 16: Tela de entrada dos dados do laudo de análise	24
Figura 17: Tela de entrada dos dados do laudo de análise preenchida.....	25
Figura 18: Tela com os resultados da interpretação com índices de fertilidade e recomendações de adubação e calagem.....	26
Figura 19: Tela inicial do aplicativo na plataforma Android e Windows Phone	27
Figura 20: Tela parcial do cadastro de culturas na plataforma Android e Windows Phone	28
Figura 21: Tela parcial de entrada dos dados do laudo de análise de solo na plataforma Android e Windows Phone	29
Figura 22: Tela parcial com os resultados da interpretação com índices de fertilidade e recomendações de adubação e calagem na plataforma Android e Windows Phone	30
Figura 23: Opiniões dos especialistas referentes ao uso do aplicativo desenvolvido	32

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
C#	C-Sharp
DTO	Data Transfer Object
ER	Entidade Relacionamento
IDE	Integrated Development Environment
K	Potássio
K ₂ O	Óxido de Potássio
LINQ	Language-Integrated Query
MVVM	Model-View-ViewModel
N	Nitrogênio
P	Fósforo
P ₂ O ₅	Pentóxido de Fósforo
pH	Potencial de Hidrogênio
PRNT	Poder Relativo de Neutralização Total
SGDB	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
TFS	Team Foundation Service
UFRGS	Universidade Federal do Rio Grande do Sul
UI	User Interface
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language

SUMÁRIO

1 INTRODUÇÃO	1
2 CONCEITOS TECNOLOGIAS E FERRAMENTAS UTILIZADOS	3
2.1 PLATAFORMA XAMARIN	3
2.2 MVVM.....	7
2.2.1 <i>View</i>	7
2.2.2 <i>ViewModel</i>	8
2.2.3 <i>Model</i>	8
2.2.4 <i>Binding</i>	8
2.2.5 <i>Exemplo de código</i>	9
2.3 SQLITE	9
2.4 FERRAMENTAS DE DESENVOLVIMENTO, CONTROLE E MODELAGEM	10
3 IMPLEMENTAÇÃO	12
3.1 PROCESSO	12
3.2 APLICATIVO	13
3.2.1 <i>Estórias de usuário</i>	14
3.2.2 <i>Modelagem das telas</i>	14
3.2.3 <i>Modelagem do banco de dados</i>	16
3.3 <i>Estrutura do Projeto</i>	17
3.4 REGRA DE NEGÓCIO	20
3.5 APRESENTAÇÃO DO APLICATIVO	21
3.5.1 <i>Testes de Execução</i>	27
3.6 LIMITAÇÕES	30
4 AVALIAÇÃO DO APLICATIVO	31
5 AVALIAÇÃO DO FRAMEWORK XAMARIN.FORMS	34
6 CONCLUSÃO E TRABALHO FUTURO	36
REFERÊNCIAS	37
ANEXO A	39
ANEXO B	41
ANEXO C	44

1 INTRODUÇÃO

O uso de tecnologias móveis tem crescido de forma muito acelerada nos últimos anos. O mercado é atualmente dominado por três plataformas: Android (Google), iOS (Apple) e Windows Phone (Microsoft). Na tentativa de atingir o maior número de usuários, os desenvolvedores de aplicativos móveis têm que atender mais de que uma única plataforma, preferencialmente todas elas. No entanto esta não é uma tarefa simples, pois essas plataformas possuem várias diferenças entre si - a começar pelo sistema operacional, linguagens de desenvolvimento, APIs (*Application Programming Interface*), locais de armazenamento, entre outros. Usualmente, desenvolver um aplicativo para estas três plataformas requer que a aplicação seja desenvolvida três vezes.

A plataforma Xamarin permite que desenvolvedores criem aplicativos multiplataforma nativos usando o mesmo código base, apenas implementando uma interface para cada sistema alvo. Como forma de reduzir ainda mais a necessidade de desenvolvimento específico de interface de usuário, um novo modelo de framework para desenvolvimento surgiu no mercado – o Xamarin.Forms – e assegura que é possível resolver esse obstáculo com apenas uma implementação.

Este trabalho visa fazer uma avaliação do Xamarin.Forms. Para isso, foi desenvolvido um aplicativo que faz a interpretação de laudos de análise de solo e gera uma recomendação de adubação e calagem (correção do índice pH do solo), para a amostra de solo em questão. Diante da complexidade e do tempo disponível, nem todas as possibilidades oferecidas pelo Xamarin.Forms foram exploradas. O aplicativo basicamente foca na comunicação de dados entre as camadas, uso de um código base único e operações de banco de dados usando SQLite. Porém, mesmo assim foi possível conhecer a plataforma e realizar uma boa avaliação de sua aplicabilidade, complexidade e utilidade.

Durante o desenvolvimento foi aplicado o MVVM (*Model-View-ViewModel*), que é um modelo já consagrado no desenvolvimento de aplicações Web orientadas pela Microsoft. Por ser um modelo baseado em UI (*User Interface*) também se adapta muito bem ao desenvolvimento de aplicativos mobile, mantendo uma clara separação entre as camadas do sistema.

Para melhorar a validação do aplicativo criado foram feitos testes com usuários especialistas, engenheiros agrônomos, esses usuários executaram um roteiro com atividades a

serem realizadas usando o aplicativo e posteriormente preencheram um questionário com uma avaliação individual.

No capítulo dois deste trabalho serão abordados os conceitos, tecnologias e ferramentas utilizadas no desenvolvimento do aplicativo. No terceiro capítulo será descrita a implementação, com seus processos, apresentação do aplicativo, regra de negócio e as limitações encontradas no desenvolvimento. No capítulo quatro estão os dados da avaliação com os usuários especialistas, compilação das respostas do questionário e mais sugestões de melhorias. Já no capítulo 5 está uma avaliação do framework Xamarin.Forms. Por fim, no capítulo 6, estão as conclusões do trabalho e mais ideias para trabalhos futuros.

2 CONCEITOS TECNOLOGIAS E FERRAMENTAS UTILIZADOS

Neste Capítulo são descritos os conceitos e as tecnologias utilizados no desenvolvimento do trabalho: Plataforma Xamarin, framework Xamarin.Forms, modelo MVVM, sistema de gerenciamento de banco de dados SQLite, IDE do Visual Studio e plataforma TFS.

2.1 Plataforma Xamarin

O Xamarin é uma plataforma que visa amenizar a necessidade de desenvolver códigos separados para cada sistema móvel existente no mercado, tornando possível criar aplicativos usando uma mesma linguagem de programação, C# (*C-Sharp*). Lembrando sempre que aplicativos criados usando Xamarin são aplicativos nativos, pois exploram todo o potencial da linguagem de desenvolvimento específica, mesmo que não seja a mesma usada pela plataforma Xamarin. Entretanto, o Xamarin não garante eliminar completamente a necessidade de desenvolvimento repetido, principalmente na camada de interface de usuário, onde ainda é necessário desenvolver código específico para cada sistema alvo. O Xamarin garante que todo o código da regra de negócio, acesso a base de dados ou comunicação com servidor seja implementada apenas uma vez (Radi, 2016) (Figura 1).

Figura 1: Esquema da plataforma Xamarin



Fonte: Adaptado de <http://www.xamarin.com/platform>

Para reduzir a necessidade de criar uma interface de usuário para cada ambiente operacional, pode-se usar o framework Xamarin.Forms, este framework torna transparente ao desenvolvedor as diferenças de cada ambiente fornecendo uma biblioteca comum. Para tornar

transparentes essas diferenças a interface é usada uma metalinguagem, no caso XAML (*Extensible Markup Language*). É importante lembrar que esta biblioteca é mais limitada que as bibliotecas específicas, tanto na questão de recursos de interface quanto ao tipo de componentes. Mas se o aplicativo a ser criado se restringir a usar os componentes e funções disponíveis torna possível a criação de uma interface única para todas as plataformas, sendo assim todo, ou praticamente todo, o código é reaproveitado. Na Figura 2 é possível verificar como o framework Xamarin.Forms se encaixa na estrutura da plataforma Xamarin.

Figura 2: Xamarin + Xamarin.Forms



Fonte: <https://www.xamarin.com/platform>

Conforme dados da empresa OPUS Software (2016), que possui 25 anos de experiência em desenvolvimento e vários projetos usando Xamarin e Xamarin.Forms, em linhas gerais para aplicativos criados usando Xamarin é possível reaproveitar em média de 40% a 60% do código (Opus Software, 2016). Já para aplicativos que usam o Xamarin.Forms, esse reaproveitamento chega na casa dos 70% a 90%, gerando um ganho significativo no processo de criação de aplicativos móveis multiplataforma (idem).

Aplicativos desenvolvidos usando Xamarin, por ter acesso total às APIs da plataforma subjacente, oferecem desempenho e usabilidade idênticos àqueles desenvolvidos utilizando ferramentas nativas dos ambientes de desenvolvimento da plataforma alvo. No caso do uso do Xamarin.Forms o desempenho em alguns casos pode sofrer algum impacto negativo (Xamarin Developer, 2016 a; MSDN Microsoft, 2016 a).

Uma grande vantagem do Xamarin está no fato de usar uma linguagem de programação única, no caso C# (*C-Sharp*). É sabido que para dominar uma linguagem e um ambiente de programação exige tempo e prática, imagine que para poder desenvolver uma aplicação o

desenvolvedor tenha que ter conhecimento da linguagem nativa e do ambiente de desenvolvimento de cada plataforma, isto traria um custo e um tempo maior no desenvolvimento da aplicação, sem contar a adaptação da equipe em cada um desses ambientes.

O uso do C# como linguagem de programação permite que o desenvolvedor utilize todos os recursos que ela oferece, é uma linguagem orientada a objetos e apresenta uma sintaxe simples e legível. Por ser integrada ao Visual Studio, a plataforma Xamarin conta com inúmeros recursos fornecidos por esta IDE (*Integrated Development Environment*) melhorando o desempenho do desenvolvedor na construção de aplicativos. Além disso, grande parte dos recursos da plataforma .Net estão disponíveis no Xamarin, plenamente integrados à linguagem C#. Podem ser citados como exemplos: LINQ, Genéricos e Listas (Xamarin, 2016).

O LINQ (*Language-Integrated Query*) oferece um modelo consistente para se trabalhar com dados em vários tipos de fontes e formatos. Usando LINQ sempre tratamos os dados como objetos e isso permite usar o mesmo padrão, independentemente da origem destes dados (MSDN Microsoft, 2016 b).

Genéricos são métodos declarados com parâmetros de tipos, tornando possível a criação de classes e métodos que adiam a especificação de um ou mais tipos até que o método ou a classe sejam declarados e instanciados pelo cliente (MSDN Microsoft, 2016 c).

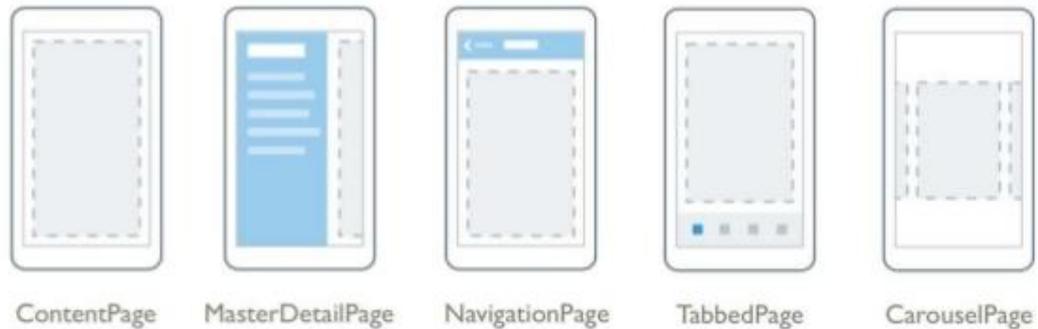
As Listas representam uma coleção de objetos fortemente tipados que podem ser acessados pelo seu índice, além de fornecer métodos para pesquisa, classificação e manipulação dos objetos nela contidos (MSDN Microsoft, 2016 d).

Os componentes da interface gráfica do Xamarin.Forms podem ser agrupados em quatro categorias: *Pages*, *Layouts*, *Views* e *Cells* (Figura 3). As *Pages* podem ser divididas em cinco diferentes tipos: *ContentPage*, *MasterDetailPage*, *NavigationPage*, *TabbedPage* e *CarouselPage* (Figura 4), cada uma delas tem apresentação e comportamentos diferentes (Xamarin Developer, 2016 b).

A *ContentPage* apresenta uma única View ou painel. Já a *MasterDetailPage* apresenta dois painéis, e oferece mais recursos para gerenciar as informações. *NavigationPage* têm a função de gerenciar a navegação entre as páginas da aplicação, cada nova página aberta é colocada em uma pilha de páginas. *TabbedPage* é um modelo de página usado para receber outras páginas dentro dela criando uma aba para cada página contida, permitindo a navegação entre elas. Por fim *CarouselPage* são similares à *TabbedPages*, porém as páginas contidas são colocadas em um carrossel, uma espécie de lista duplamente encadeada circular. A escolha do

tipo de Page a ser usada está a cargo da necessidade da aplicação, de qual forma os dados ficam melhor dispostos ao usuário final (Xamarin Developer, 2016 c).

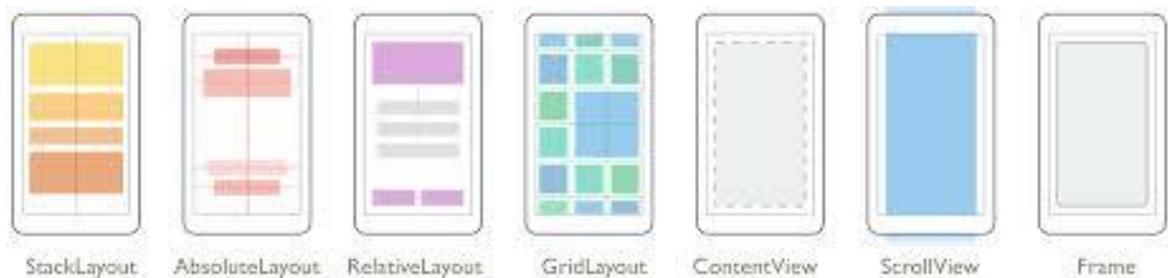
Figura 3: Tipos de Pages mais utilizadas



Fonte: <https://developer.xamarin.com/guides/xamarin-forms/controls/Pages/>

Para o posicionamento dos componentes dentro da página existem diferentes tipos de Layouts que podem ser usados conforme a necessidade do desenvolvedor. Na Figura 4 pode-se ver a aparência dos principais Layouts fornecidos pelo Xamarin.Forms (Xamarin Developer, 2016 d).

Figura 4: Tipo de Layouts mais utilizados



Fonte: <https://developer.xamarin.com/guides/xamarin-forms/controls/layouts/>

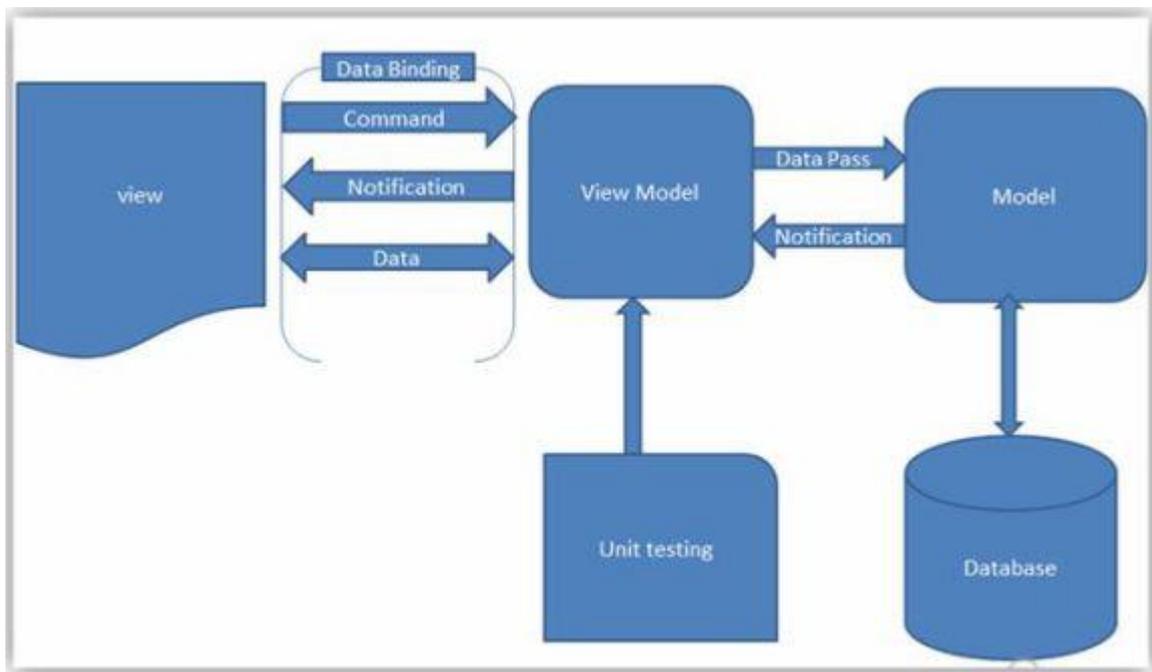
As Views no Xamarin.Forms são os objetos visuais da tela, como botões, labels, caixas de entrada de texto, comumente conhecidos como componentes de interface de usuário. Cells são elementos não visuais, porém são usados para modelar a apresentação visual de outros itens na tela.

Os componentes da interface gráfica podem ser definidos de duas formas, usando C# ou XAML que é o mais indicado para o uso do padrão de projeto MVVM.

2.2 MVVM

MVVM foi criado em 2005 por John Gossman, um dos arquitetos do WPF (*Windows Presentation Foundation*) e Silverlight. É um padrão de projeto de software baseado em UI. Este padrão visa uma clara separação de responsabilidades entre as camadas do sistema. A comunicação entre a camada de interface e a ViewModel se dá através do mecanismo Binding, como pode ser vista na Figura 5. Essa separação entre camadas permite que o código tenha algumas características desejáveis como: testabilidade, modularidade, manutenibilidade e flexibilidade (MSDN Microsoft, 2012 e; Devmedia, 2014).

Figura 5: Esquemático da arquitetura MVVM



Fonte: <http://netcoders.com.br/blog/introducao-ao-mvvm/>

Os principais componentes do modelo MVVM são: Model, View e ViewModel, abaixo está a definição e responsabilidade de cada componente.

2.2.1 View

A View representa a interface de usuário, pode ser uma tela, uma janela ou uma interface qualquer de entrada e saída de dados. A View não executa nenhuma operação no sistema, ficando responsável apenas por apresentar as informações na tela e receber os comandos do usuário.

A ligação entre a View e a ViewModel se dá através da propriedade `BindingContext` da View (Figura 6). Esta propriedade armazena uma referência da `ViewModel` que através da propriedade `Binding` se comunica com a `ViewModel`, passando e recebendo dados, enviando comandos e recebendo notificações da `ViewModel` (Figura 5).

Figura 6: Exemplo de utilização da propriedade `BindingContext`

```
namespace TCC.Views
{
    public partial class MainView : ContentPage
    {
        public MainView()
        {
            InitializeComponent();
            this.BindingContext = new MainViewModel();
        }
    }
}
```

Fonte: Próprio autor

2.2.2 ViewModel

A `ViewModel` é a classe que fornece os dados e executa os comandos que a `View` necessita. Ela não possui nenhuma referência com a `View`, sua comunicação se dá através da exposição de suas propriedades e do *Binding* feito pela `View`. É dessa forma que a `View` é notificada das alterações nos dados da `ViewModel` e também que a `ViewModel` recebe as modificações feitas na `View`.

A `ViewModel` se conecta diretamente com a `Model`, tendo acesso a todas as propriedades e funcionalidades disponibilizadas pelo modelo.

2.2.3 Model

A `Model` é a classe que contém a definição do modelo dos dados, propriedades e métodos. É comum que essa classe faça a validação dos dados de entrada vindo da `View`.

2.2.4 Binding

Binding é o mecanismo que uma `View` usa para se comunicar com a `ViewModel`, é através dessa associação que os dados da `View` refletem em alterações nas propriedades da `ViewModel` e que a `View` é notificada das alterações das propriedades da `ViewModel`.

2.2.5 Exemplo de código

2.2.5.1 View

```
<TextBox Text="{Binding Nome}"/>
```

2.2.5.2 ViewModel

```
public string Nome
{
    get
    {
        return _nome;
    }
    set
    {
        _nome = value;
        this.Notify("Nome");
    }
}
```

Vale lembrar que é a View que faz o Binding e ele pode apresentar quatro diferentes formatos:

- *Default*: Segue o sentido do fluxo de informações, que é dependente do componente em questão. Por exemplo: Label - OneWay, TextBox - TwoWay.
- *OneWay*: Único sentido, alterações feitas na fonte são transmitidas para o destino.
- *OneTime*: O fluxo de dados é semelhante ao OneWay, porém só acontece na inicialização do componente. Mesmo que os dados sejam alterados posteriormente na fonte, as alterações não são refletidas para o destino.
- *TwoWay*: Nesse modo toda a alteração realizada em um lado é refletida no outro.

2.3 SQLite

SQLite é uma biblioteca escrita em linguagem C que implementa um mecanismo de banco de dados SQL autônomo. Programas que usam a biblioteca SQLite podem ter acesso aos dados sem executar um processo SGDB (*Sistema de Gerenciamento de Banco de Dados*) separado, lendo e escrevendo no próprio arquivo do banco de dados no disco local, não há necessidade de um servidor. É um sistema completo de Tabelas, Índices, Triggers e Views contido em apenas um arquivo, e esse arquivo tem formato multiplataforma sendo possível ser copiado entre diferentes tipos de sistemas. Essas características fazem com que o SQLite seja um dos modelos de base de dados mais usados no mundo.

O uso do SQLite é recomendado onde a simplicidade da implementação e manutenção são mais importantes que incontáveis recursos de SGDB exigidos por aplicações mais complexas, Situações onde a simplicidade é a melhor escolha (SQLite).

2.4 Ferramentas de desenvolvimento, controle e modelagem

O Visual Studio 2015 Community foi a IDE de desenvolvimento utilizada para criação do aplicativo. Esta versão é gratuita para estudantes e traz uma integração nativa com a plataforma Xamarin e com a linguagem C#, além de ser uma boa IDE de desenvolvimento, devido aos inúmeros recursos e facilidades oferecidas ao desenvolvedor (MSDN Microsoft, 2016 f).

Além de fornecer uma ferramenta para a escrita de código, o Visual Studio também oferece integração com o TFS (*Team Foundation Service*), que é uma plataforma de serviços online da Microsoft que oferece serviços de Versionamento de código, testes automatizados, integração contínua e um módulo para gerenciamento de projetos. Dentre todas essas possibilidades, foram usadas as funções de versionamento de código e gerenciador do projeto (TFS Microsoft, 2016).

A modelagem da interface de usuário foi criada usando o aplicativo Balsamiq Mockups 3, o qual é simples de usar e facilita o processo de criação de protótipos de interface gráfica, onde em minutos é possível criar uma tela completa. Ele pode ser usado para modelar qualquer tipo de interface desktop, móvel ou Web, fornecendo todos os tipos de controles existentes em qualquer plataforma (Balsamiq Mockups, 2016).

Para criação da base de dados e entidades no banco foi usado o mapeamento através das classes do modelo. O SQLite se encarrega dessa responsabilidade, basta apenas fazer a chamada de uma função genérica, passando por parâmetro o tipo do objeto que vai corresponder a uma entidade no banco. Também é possível fazer essa criação passando o comando em formato de texto estruturado na linguagem SQL (*Structured Query Language*), mas, pela praticidade do primeiro método, esse é que foi adotado no trabalho (Felix, 2016).

Devido à simplicidade da base de dados para criação do modelo ER, uma ferramenta online, chamada Ondras, foi utilizada. Essa ferramenta pode ser encontrada no site <http://ondras.zarovi.cz/sql/demo/>, e ela permite que sejam criadas as entidades e relacionamentos entre as entidades. Ela peca apenas em não apresentar no formato gráfico qual

é a forma do relacionamento, se 1..1 (um para um), 1..n (um para muitos) ou n..n (muitos para muitos).

3 IMPLEMENTAÇÃO

Neste Capítulo são descritos os detalhes da implementação do aplicativo: Processo de desenvolvimento adotado, apresentação do aplicativo criado, estórias de usuário, finalidade e objetivos, modelagens de interface de usuário e banco de dados, estrutura do projeto, entrada e saída de dados, regra de negócio, dificuldades e limitações.

3.1 Processo

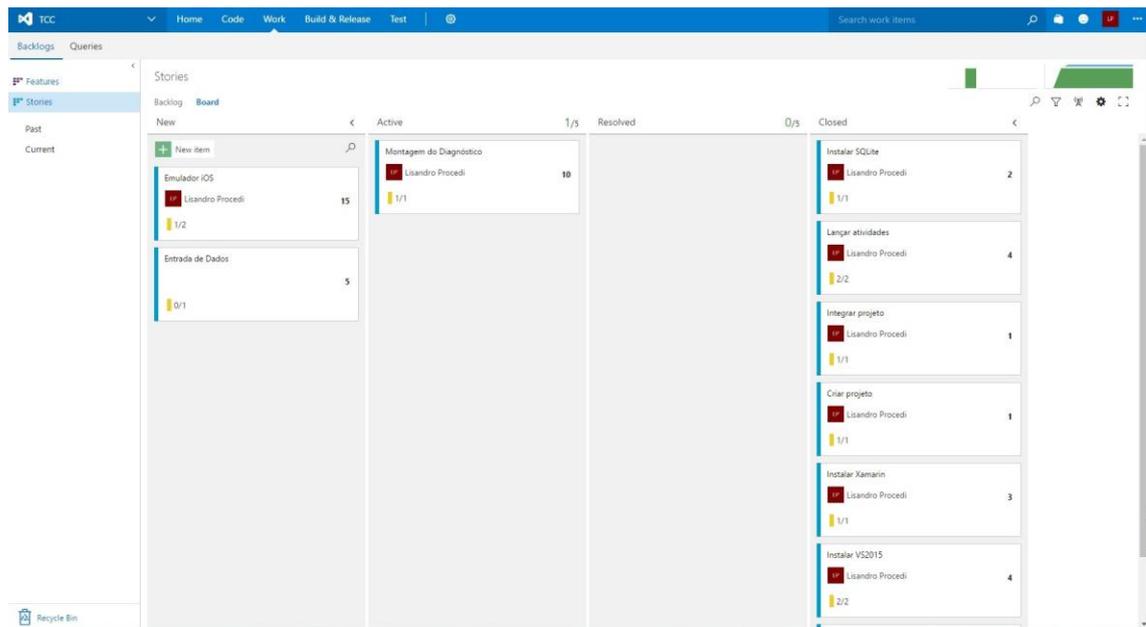
O processo de desenvolvimento seguiu uma abordagem ágil híbrida entre Kanban e Scrum. Kanban segue o conceito de utilização de cartões (*post-it* e outros) afim de indicar o andamento do fluxo das atividades no desenvolvimento de softwares, teve origem na indústria automobilística para controlar a produção em chão de fábrica. Scrum é a ideia de entregar o máximo de valor (resultados) ao cliente no menor tempo possível (Kniberg & Skarin, 2010).

Pelo fato de existir apenas uma pessoa na equipe foi necessário abolir algumas partes do processo ágil, por exemplo a reunião diária. Com o uso da plataforma TFS foi possível organizar as tarefas na forma de Kanban (Figura 7), além de atribuir importância e tempo previsto de desenvolvimento para cada uma delas. Essa ferramenta oferece recursos para que o desenvolvedor acompanhe o andamento das atividades como: *Burndown Chart*¹ e indicadores da velocidade de andamento do trabalho, permitindo que a cada iteração seja ajustada a quantidade de atividades selecionadas.

As iterações foram divididas em semanas, pois esta era a frequência das reuniões com o orientador, ao final de cada semana uma nova iteração era planejada. As estórias de usuário utilizadas para criação das tarefas foram criadas com ajuda dos agrônomos da Faculdade de Agronomia da UFRGS (*Universidade Federal do Rio Grande do Sul*), as mesmas pessoas que posteriormente fizeram a avaliação final do aplicativo.

¹ Burndown Chart é uma representação gráfica do Trabalho X Tempo. O trabalho pendente é representado no eixo vertical e o tempo na horizontal. É um gráfico útil para apresentar o andamento do trabalho gerando uma previsão de quando o trabalho será concluído (Wenzel, 2010).

Figura 7: Tela do TFS com as atividades de uma iteração



Fonte: Próprio autor

3.2 Aplicativo

Antes do início de cada safra, é comum que os agricultores, com a ajuda de um engenheiro agrônomo, façam a coleta de amostras de solo da área que será cultivada. Estas amostras são encaminhadas para o Laboratório de Análises de Solo da Faculdade de Agronomia da UFRGS, onde por processos físicos e químicos, feitos de acordo com métodos oficiais de análise de solo, é gerando um Laudo de Análise (Figura 12). Com base nos dados do laudo e através das tabelas e fórmulas contidas no Manual de Adubação e Calagem (Rolas, 2004) o engenheiro agrônomo calcula a recomendação de adubação e calagem para a área amostrada.

O aplicativo desenvolvido tem por finalidade interpretar esses laudos de análise e gerar os índices de fertilidade e a recomendação de adubação e calagem para uma determinada amostra. A motivação para desenvolvimento do aplicativo surgiu, quando em conversa com engenheiros agrônomos, revelaram que esse processo de cálculo para definir a recomendação ainda é feito de forma manual para todas as amostras.

O objetivo principal do aplicativo é facilitar o trabalho do engenheiro agrônomo na interpretação de laudos e definição da recomendação de adubação e calagem. Com apoio do aplicativo essa tarefa se torna mais rápida e simples, e pode evitar erros na manipulação dos dados, uma vez que é necessário apenas entrar com as informações do laudo de análise e o

sistema gera os resultados. Além disso por ser um aplicativo móvel permite que ele seja usado em qualquer local, visto que normalmente o engenheiro agrônomo trabalha no campo.

As funcionalidades oferecidas pelo sistema são o cadastro de culturas e entrada dos dados do laudo de análise. Com base nessas informações o sistema está apto a gerar os índices de fertilidade e recomendações de adubação e calagem.

3.2.1 Estórias de usuário

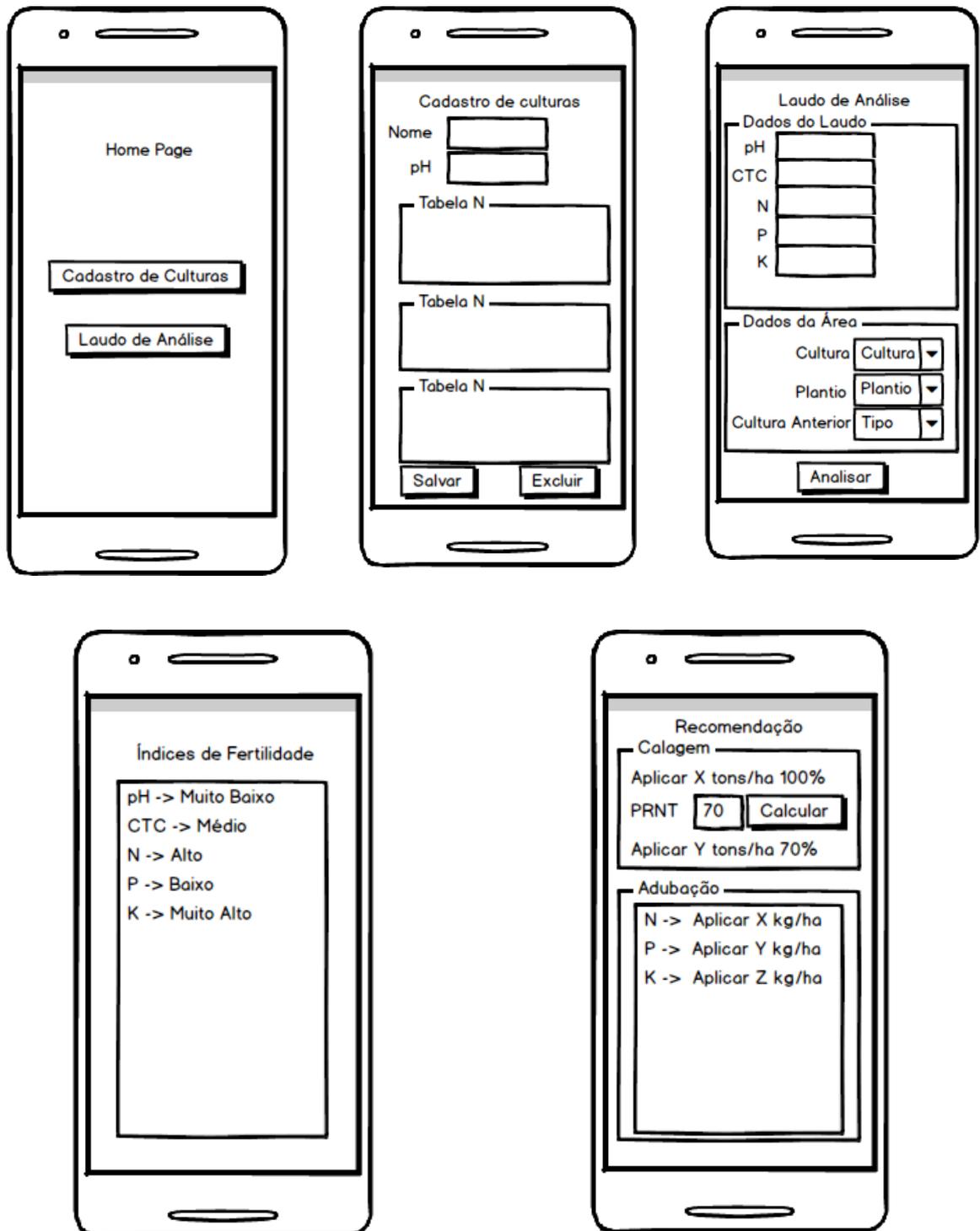
Abaixo são apresentadas as estórias de usuário que foram criadas juntamente com os engenheiros agrônomos e que serviram de base para o desenvolvimento do aplicativo:

- Sendo um engenheiro agrônomo e tendo em mãos o laudo de análise quero entrar com os dados do laudo para que o sistema me retorne um diagnóstico simplificado dos índices de fertilidade da amostra.
- Sendo um engenheiro agrônomo quero poder cadastrar as culturas com suas tabelas de N (*Nitrogênio*), P (*Fósforo*) e K (*Potássio*) e faixa ideal de pH (*Potencial de Hidrogênio*) para que o sistema use estes dados como base nos cálculos de recomendação e calagem.
- Sendo um engenheiro agrônomo e tendo informado ao sistema os dados do laudo de análise e a cultura desejada quero que o sistema me recomende qual calagem deve ser aplicada
- Sendo um engenheiro agrônomo e tendo a recomendação da calagem quero que o sistema me permita entrar com os dados do PRNT (*Poder Relativo de Neutralização Total*) do calcário comercial para que o valor real da quantidade a ser aplicada seja gerado.
- Sendo um engenheiro agrônomo e tendo informado os dados do laudo de análise, qual a cultura a ser introduzida, cultura anterior e qual o cultivo quero que o sistema calcule as quantidades de N, P e K a serem aplicadas.

3.2.2 Modelagem das telas

Baseado nas estórias de usuário e conversas com os especialistas (engenheiros agrônomos) foi possível criar uma modelagem da interface gráfica. Foram definidas as telas base que o aplicativo deveria possuir. Na Figura 8 se encontram os protótipos das telas que foram criadas para o aplicativo usando o software Balsamiq Mockups 3.

Figura 8: Protótipos de interface de usuário



Fonte: Próprio autor

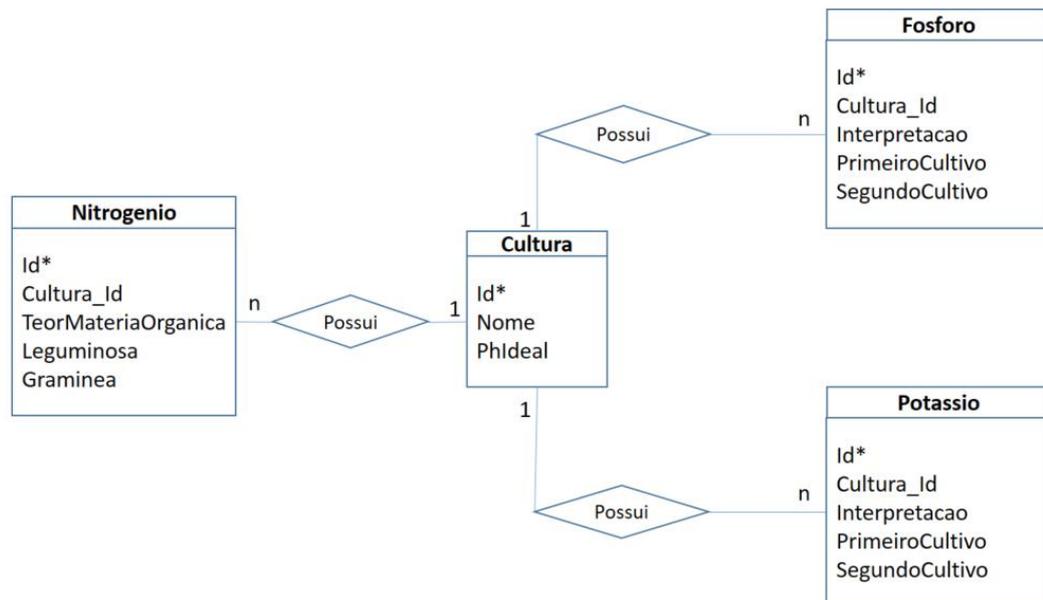
Na figura anterior encontram-se as *mockups* das telas do aplicativo. Na primeira parte está a tela de entrada com suas duas opções de entrada. Em seguida vem o cadastro de culturas, contendo as informações básicas da cultura e mais as tabelas de N, P e K. No final da primeira

parte encontra-se a tela de entrada dos dados do laudo de análise e os dados da área a ser cultivada. Na segunda parte aparece a tela com a informação dos índices de fertilidade e, para concluir, vem a tela da recomendação de adubação e calagem, na parte da calagem está o campo para entrada do valor de PRNT para ajuste do valor total.

3.2.3 Modelagem do banco de dados

A modelagem do banco de dados aconteceu de forma natural conforme o mapeamento das classes no sistema. Tem-se apenas um cadastro para este aplicativo então a base de dados se resumiu a uma entidade Cultura que armazena os dados básicos e mais três tabelas auxiliares que armazenam as listas de N, P e K relacionados a essa cultura. Os relacionamentos da entidade Cultura com as demais entidades é do tipo 1..n (um para muitos). Na Figura 9 tem-se uma visualização do modelo ER (*Entidade Relacionamento*) para este aplicativo.

Figura 9: Modelo ER da base de dados



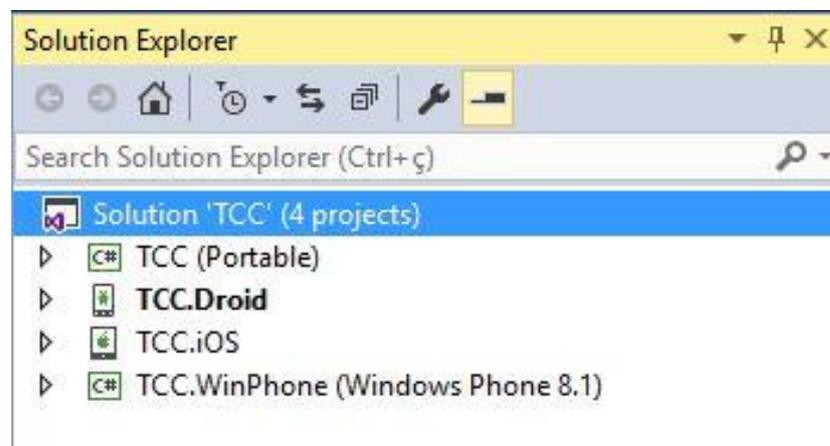
Fonte: Próprio autor

Pode-se perceber 4 entidades (Cultura, Nitrogênio, Fósforo e Potássio) e seus atributos. A entidade Cultura possui uma lista para cada um dos nutrientes N, P e K. Os atributos sinalizados com asterisco são chaves primárias da entidade.

3.3 Estrutura do Projeto

Quando um novo projeto do tipo multiplataforma (*cross-platform*) é criado usando o Visual Studio, a sua estrutura básica é como a mostrada na Figura 10, onde se tem uma *Solution* que contém quatro projetos dentro dela. O primeiro projeto (do tipo *Portable*) é o que contém o código compartilhado por todos os projetos da *Solution*, e é neste projeto que devemos implementar tudo o que é comum, código base, independentemente da plataforma móvel que vai rodar o aplicativo final. Os três projetos restantes são específicos de cada plataforma suportada pelo Xamarin, respectivamente Android, iOS e Windows Phone, nesses projetos só deve ser implementado o código específico da plataforma, e que irá complementar o código base (compartilhado).

Figura 10: Representação da estrutura básica do projeto

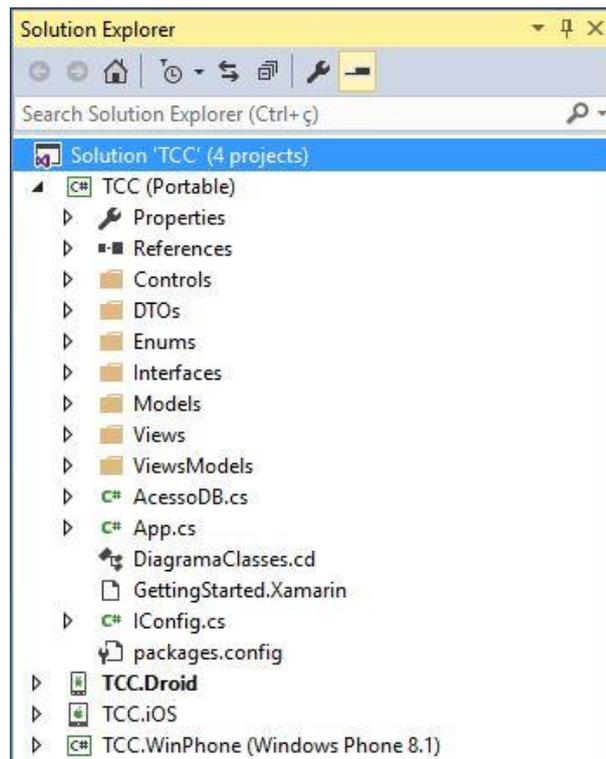


Fonte: Próprio autor

Como optou-se pelo uso do framework Xamarin.Forms e o aplicativo em questão é bem simples, a quantidade de código específico implementado se reduziu a encontrar o caminho da base de dados para cada uma das plataformas. Para facilitar ainda mais o trabalho, uma Interface foi criada no código compartilhado e implementada nos projetos específicos, dessa maneira a aplicação resolve o caminho em tempo de execução apenas fazendo uma chamada dos métodos da interface.

O projeto compartilhado foi dividido em diretórios conforme as camadas do sistema, View, ViewModel e Model, como pode ser visto na Figura 11, o que dessa forma facilita o entendimento do código por qualquer desenvolvedor que por ventura venha a trabalhar nesse projeto.

Figura 11: Divisão da estrutura do projeto conforme modelo MVVM



Fonte: Próprio autor

3.3.1 Entrada e Saída de dados

Por ser um aplicativo com uma finalidade bem específica, sua entrada de dados se resume a um cadastro de culturas e entrada das informações do laudo de análise (Figura 12). Todos os dados dos formulários devem ser preenchidos pois é baseado nessas entradas que o sistema, através de seu algoritmo, gera as informações de saída.

Figura 12: Laudo de análise de solo

FACULDADE DE AGRONOMIA - DEPTO. DE SOLOS
LABORATÓRIO DE ANÁLISES

49 anos
Servindo a Agricultura

Laudo de Análise de Solo

NOME: _____
MUNICÍPIO: SÃO FRANCISCO DE PAULA
ESTADO: RS
LOCALIDADE: _____

DATA DO RECEBIMENTO: 06/10/16
DATA DA EXPEDIÇÃO: 17/10/16

NUM	REGISTRO	ARGILA %	pH H ₂ O	Índice SMP	P mg/dm ³	K mg/dm ³	M.O. %	Al _{troc.} cmol _c /dm ³	Ca _{troc.} cmol _c /dm ³	Mg _{troc.} cmol _c /dm ³
1	606/35	13	4.9	5.0	4.1	48	5.1	1.9	4.0	1.2
2	606/36	13	4.8	4.6	2.7	40	8.2	4.1	0.6	0.2
3	606/37	12	5.0	5.0	28	252	9.9	1.0	8.2	2.6
4	606/38	8	4.5	4.6	6.9	71	9.2	2.9	1.9	0.9
5	606/39	9	5.3	5.3	23	291	>10	0.4	9.6	4.7

Argila determinada pelo método do densímetro; pH em água 1:1; P, K, Cu, Zn e Na determinados pelo método Mehlich 1; M.O. por digestão úmida; Ca, Mg, Al e Mn trocáveis extraídos com KCl 1 mol L⁻¹; S-SO₄ extraído com CaHPO₄ 500 mg L⁻¹ de P; B extraído com água quente.

NUM	Al+H cmol _c /dm ³	CTC cmol _c /dm ³	% SAT da CTC		RELAÇÕES			SUGESTÃO DE CALAGEM p/PRNT (t ha ⁻¹)			
			BASES	Al	Ca/Mg	Ca/K	Mg/K	100	85	70	55
1	13.7	19.1	28	26.3	3.3	33	10				
2	21.8	22.7	4	81.8	3.0	6	2.0				
3	13.7	25.2	45	8.0	3.2	13	4.0				
4	21.8	24.7	12	49.2	2.1	10	4.9				
5	9.7	24.8	61	2.6	2.0	13	6				

CTC a pH 7.0. Necessidade de calcário para atingir pH 6,0 - calculada pela média dos métodos SMP e Al+MO. Sugestão válida no caso de não ter sido feita calagem integral nos últimos 3 anos e sob sistema de cultivo convencional. No sistema plantio direto, consultar um agrônomo.

NUM	S mg/dm ³	Zn mg/dm ³	Cu mg/dm ³	B mg/dm ³	Mn mg/dm ³	Fe g/dm ³	Na mg/dm ³	OUTRAS DETERMINAÇÕES
1	5.5	5.4	2.2	0.5	8			
2	5.8	0.9	0.7	0.8	6			
3	12	2.5	0.4	1.0	3			
4	17	1.4	0.6	1.0	4			
5	8.1	1.7	0.2	1.0	2			

Consulte um agrônomo para obter as recomendações de adubação

NUM	IDENTIFICAÇÃO DA AMOSTRA
1	01 - BAIXADA DA CASA - 0,15ha - TIFTON
2	02 - GASODUTO - 0,1ha - PASTAGEM
3	03 - ENO LAVOURINHA - 1,5ha - PASTAGEM
4	04 - LAVOURA JORGINHO - 6ha - PASTAGEM
5	05 - LAVOURA LINDEIRA - 4ha - PASTAGEM

Clesio Gianello
Eng^o Ag^o CREA 8^o Reg 25.642
Chefe do Laboratório de Análises

Laboratório de Análises de Solo - Av. Bento Gonçalves, 7712 - Porto Alegre - RS - CEP 91540-000
Fones/Fax: (0xx51) 3308-6023 - 3308-7457 - E-mail: lsolos@hotmail.com - www.ufrgs.br/labsolos

Fonte: Laboratório de análise de solos da Faculdade de Agronomia da UFRGS

São geradas três informações de saída: índice de fertilidade da amostra, recomendação de calagem e recomendação de adubação. Os índices de fertilidade são importantes para situar o engenheiro agrônomo de como estão as componentes da amostra analisada, baseado nesses índices é possível que alguma das recomendações de adubação e calagem sofra alguns ajustes,

feitos pelo especialista. A recomendação de correção do pH do solo (calagem) é gerada considerando um PRNT de 100%, porém isso não ocorre comercialmente onde as fórmulas variam de 60% a 90% (dados informados pelos engenheiros agrônomos), sendo assim é necessário informar ao sistema qual fórmula comercial será usada para ser possível gerar a recomendação corretamente. A recomendação de adubação é gerada considerando aplicações de N, P₂O₅ (*Pentóxido de Fósforo*) e K₂O (*Óxido de Potássio*) a 100%, isso também não ocorre comercialmente, porém com a adubação o fator de correção é mais complexo uma vez que as fórmulas geralmente são mistas e não usam exatamente estes componentes.

Todos os campos de entrada do sistema recebem uma validação, esta validação é realizada em dois momentos: primeiramente no componente da interface de usuário onde é validada a entrada conforme o tipo do campo, isso antecipa erros de digitação uma vez que dados não permitidos já são filtrados. Em um segundo momento é feita a validação do dado recebido da interface gráfica pela ViewModel, essa segunda validação é feita na Model, dessa forma é garantida a integridade dos dados do sistema.

3.4 Regra de negócio

O desenvolvimento da regra de negócio do sistema se baseou no Manual de Adubação e Calagem, esse manual é que define como devem ser interpretados os dados do laudo de análise para recomendações de adubação e calagem.

Para montar os índices de fertilidade de uma amostra são usadas as tabelas 5.1 a 5.5, as mesmas estão no Anexo B. Para cada um dos índices foi criado um método no sistema que recebe o valor proveniente do laudo e enquadra esse valor em uma categoria conforme escala da tabela em questão.

Semelhante aos índices de fertilidade, a recomendação de calagem é baseada na tabela 6.2 do manual (Anexo B). Existe um método que recebe o pH ideal para a cultura informada e também o índice SMP contido no laudo, baseado nessas informações o sistema define qual deve ser a quantidade de calcário a ser aplicada, em valor 100%, a correção dessa quantidade é feita posteriormente para percentuais de formulações comerciais.

Por fim, para definir a recomendação de adubação, o sistema usa os dados cadastrados para a cultura. Para o caso do Nitrogênio é necessário analisar teores de matéria orgânica e qual tipo de cultura estava presente no plantio anterior, já a recomendação de Fósforo e Potássio é

Figura 14: Tela do cadastro de culturas

The figure displays four screenshots of a mobile application interface for crop registration. Each screenshot shows a dark-themed screen with a blue header containing a back arrow and a logo with the letter 'X'. The status bar at the top indicates 69% battery and 12:02.

Screenshot 1 (Top Left): Shows the main registration screen. The title is "Cadastro de Culturas". Below it, the section "Dados da Cultura" contains a text input field for "Nome" (pre-filled with "Nome da Cultura") and a numeric input field for "pH Ideal" (pre-filled with "0.0").

Screenshot 2 (Top Right): Shows the "Nitrogênio" (Nitrogen) configuration screen. The title is "Cadastro de Culturas - Nitrogênio". It specifies "kg de N/ha". The "Teor de Matéria Orgânica %" is set to "<2,5". The "Cultura Anterior" section has two columns: "Leguminosa" and "Gramínea", each with a numeric input field.

Screenshot 3 (Bottom Left): Shows the "Fósforo" (Phosphorus) configuration screen. The title is "Cadastro de Culturas - Fósforo". It specifies "kg de P2O5/ha". The "Teor de P" is set to "Muito Baixo". The "Fósforo por Cultivo" section has two columns: "1º Cultivo" and "2º Cultivo", each with a numeric input field.

Screenshot 4 (Bottom Right): Shows the "Potássio" (Potassium) configuration screen. The title is "Cadastro de Culturas - Potássio". It specifies "kg de K2O/ha". The "Teor de K" is set to "Muito Baixo". The "Potássio por Cultivo" section has two columns: "1º Cultivo" and "2º Cultivo", each with a numeric input field. At the bottom, there are three buttons: "SALVAR", "EXCLUIR", and "LIMPAR".

Fonte: Próprio autor

Figura 15: Tela do cadastro de culturas preenchida

Nome **Trigo**

pH Ideal **5.8**

kg de N/ha

Teor de Matéria Orgânica %	Cultura Anterior	
	Leguminosa	Gramínea
<2,5	80	100
2,5 - 5,0	60	80
> 5,0	40	60

kg de P2O5/ha

Teor de P	Fósforo por Cultivo	
	1º Cultivo	2º Cultivo
Muito Baixo	120	140
Baixo	100	120
Médio	80	100
Alto	60	80
Muito Alto	40	60

kg de K2O/ha

Teor de K	Potássio por Cultivo	
	1º Cultivo	2º Cultivo
Muito Baixo	100	110
Baixo	90	100
Médio	80	90
Alto	60	70
Muito Alto	30	50

SALVAR EXCLUIR LIMPAR

Fonte: Próprio autor

Se optarmos por escolher o segundo botão (Laudo de Análise) na tela inicial será apresentado o formulário para lançamento das informações contidas no laudo de análise (Figura 16). Além dos dados do laudo também é necessário informar qual a cultura será cultivada na área da amostra, essa cultura é proveniente do cadastro de culturas. Também deve ser informado qual tipo de cultura estava presente na área no período anterior e se é o primeiro plantio dessa cultura na área. Na Figura 17 podemos ver um exemplo deste formulário preenchido. Nesta tela temos a opção de: Analisar os dados e gerar os resultados ou Limpar os campos da tela.

Figura 16: Tela de entrada dos dados do laudo de análise

The image displays two screenshots of a mobile application interface for data entry. The left screenshot shows a scrollable list of input fields for soil analysis data. The right screenshot shows the same fields with a 'Dados da Cultura' section added at the bottom, including 'Cultura', 'Cultura Anterior', and 'Cultivo' fields, and two buttons: 'ANALISAR' and 'LIMPAR'.

Dados do Laudo de Análise

Teor de Argila % argila

pH (H2O) pH em água

SMP Índice SMP

Fósforo Teor P

Potássio Teor K

Matéria Orgânica Teor de M.O.

CTC Índice CTC

Dados da Cultura

Cultura

Cultura Anterior

Cultivo

ANALISAR LIMPAR

Fonte: Próprio autor

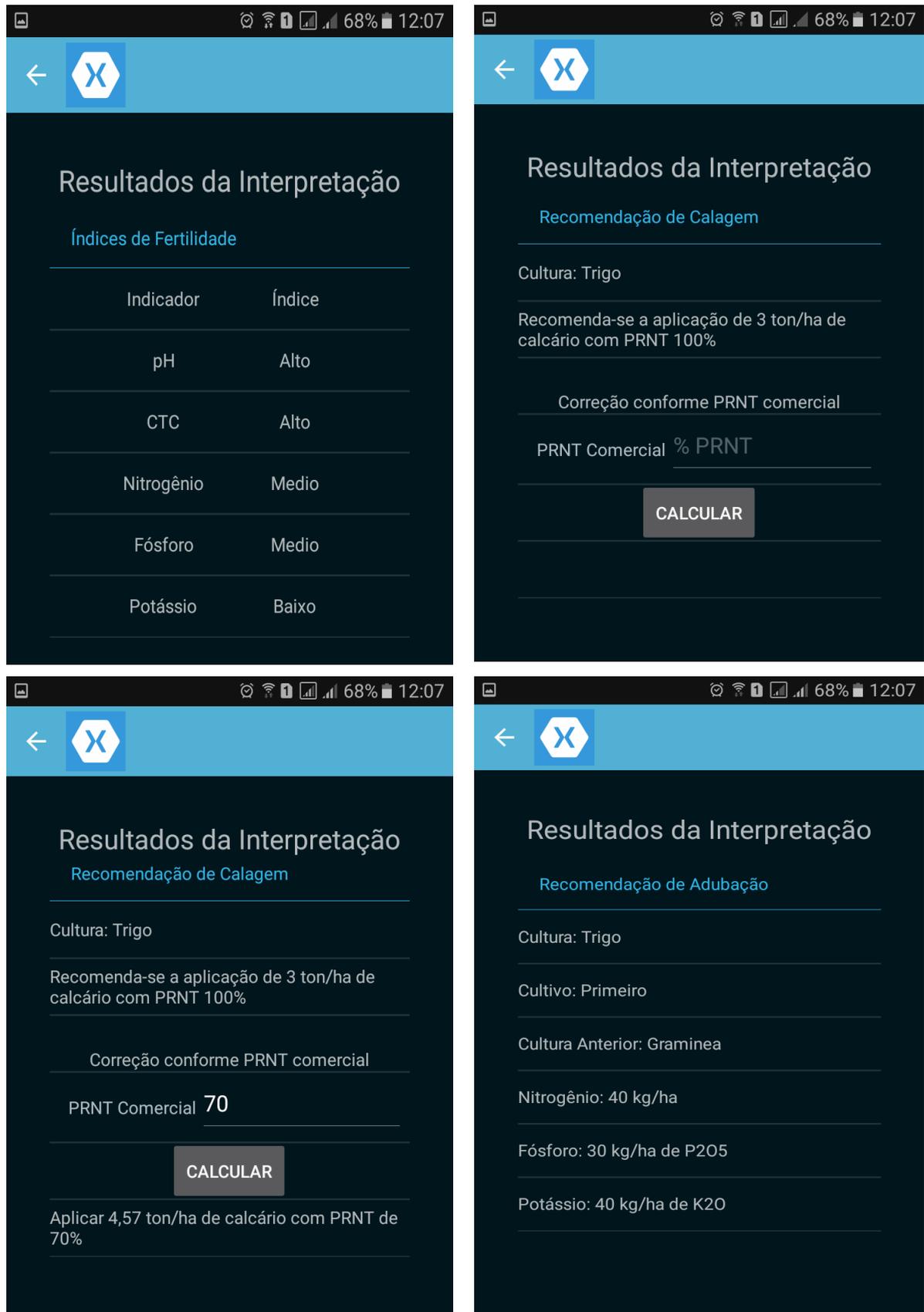
Figura 17: Tela de entrada dos dados do laudo de análise preenchida

The image displays two screenshots of a mobile application interface. The left screenshot shows the 'Dados do Laudo de Análise' (Soil Analysis Report Data) section, which is a form with the following filled-in fields: Teor de Argila (35), pH (H2O) (5.8), SMP (6), Fósforo (10), Potássio (55), Matéria Orgânica (3), and CTC (21.7). The right screenshot shows the 'Dados da Cultura' (Cultural Data) section, which is a form with the following filled-in fields: Trigo, Gramínea, and Primeiro. At the bottom of the right screenshot, there are two buttons: ANALISAR and LIMPAR.

Fonte: Próprio autor

Após clicar em Analisar na tela de entrada dos dados do laudo, o sistema usa as regras implementadas para gerar os resultados, que são apresentados em uma nova página como pode ser visto na Figura 18. Esta tela tem três seções, a primeira contém os índices de fertilidade que servem de indicativo para o engenheiro agrônomo saber como está este solo. Na segunda seção temos a indicação de calagem, nessa parte é necessário informar os dados do PRNT da fórmula comercial do calcário a ser aplicado para que o valor correto seja calculado. E por fim são apresentados os dados de adubação, com a indicação das quantidades de N, P e K a serem aplicadas.

Figura 18: Tela com os resultados da interpretação com índices de fertilidade e recomendações de adubação e calagem

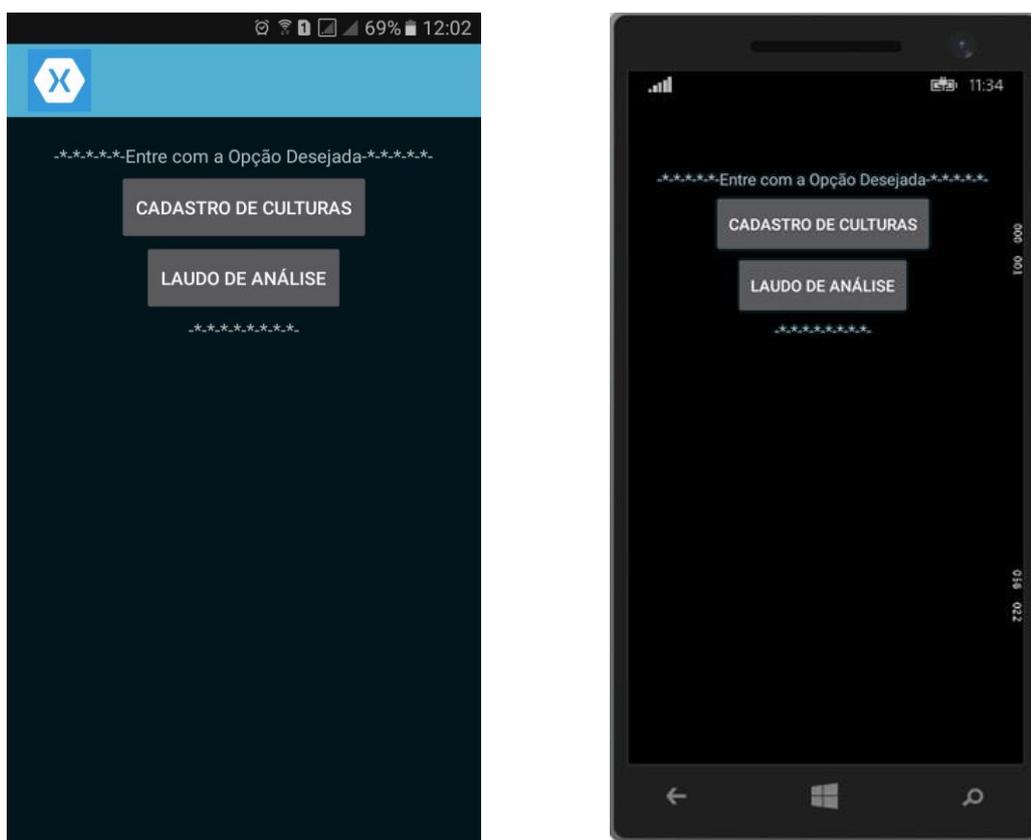


Fonte: Próprio autor

3.5.1 Testes de Execução

Durante o desenvolvimento do aplicativo foram feitos testes de execução para todas as telas e funções criadas. Essas execuções foram feitas nas plataformas Android e Windows Phone usando o mesmo código fonte e usando os emuladores das plataformas. As execuções eram disparadas simultaneamente usando a opção *Multiple Startup Projects*, no Visual Studio. A seguir temos exemplos dos resultados obtidos durante as execuções.

Figura 19: Tela inicial do aplicativo na plataforma Android e Windows Phone



Fonte: Próprio autor

Figura 20: Tela parcial do cadastro de culturas na plataforma Android e Windows Phone

The figure displays two side-by-side screenshots of a mobile application interface for crop registration. The left screenshot is for an Android device, showing a dark-themed screen with a blue header bar containing a back arrow and a logo. The right screenshot is for a Windows Phone device, showing a light-themed screen with a blue header bar containing a back arrow, the Windows logo, and a search icon.

Both screens display the following form fields and sections:

- Header:** "Cadastro de Culturas"
- Dados da Cultura:**
 - Nome: Nome da Cultura
 - pH Ideal: 0,0
- Nitrogênio:**
 - kg de N/ha
 - Teor de Matéria Orgânica %
 - Cultura Anterior: Leguminosa, Gramínea
 - Options: <2,5, 2,5 - 5,0, > 5,0
- Fósforo:**
 - kg de P2O5/ha
 - Teor de P
 - Fósforo por Cultivo

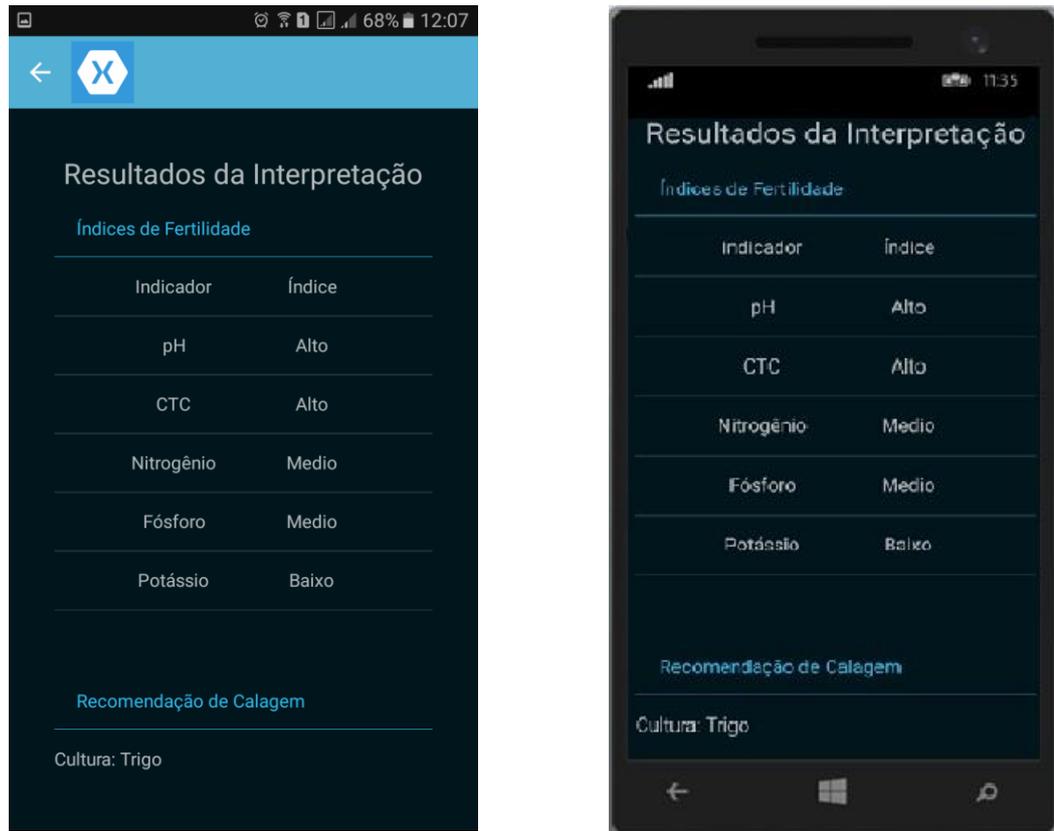
Fonte: Próprio autor

Figura 21: Tela parcial de entrada dos dados do laudo de análise de solo na plataforma Android e Windows Phone

The image displays two mobile application screens for soil analysis data entry. The left screen is for Android, featuring a blue header with a back arrow and a logo. Below the header, the text 'Dados do Laudo de Análise' is displayed. The form contains several input fields with labels: 'Teor de Argila % argila', 'pH (H2O) pH em água', 'SMP Índice SMP', 'Fósforo Teor P', 'Potássio Teor K', 'Matéria Orgânica Teor de M.O.', and 'CTC Índice CTC'. At the bottom, there is a section for 'Dados da Cultura' and two buttons: 'ANALISAR' and 'LIMPAR'. The right screen is for Windows Phone, showing a dark theme. It has a similar layout with the same input fields and buttons. The status bar at the top of the Windows Phone screen shows the time as 11:35.

Fonte: Próprio Autor

Figura 22: Tela parcial com os resultados da interpretação com índices de fertilidade e recomendações de adubação e calagem na plataforma Android e Windows Phone



Fonte: Próprio autor

Como pode ser visto nas imagens 19 a 22 o resultado em ambas as plataformas é exatamente o mesmo.

3.6 Limitações

A principal limitação do projeto se resumiu ao uso da plataforma iOS, ou o não uso dela. Por não possuir um dispositivo móvel da Apple não foi possível fazer testes usando este sistema, uma vez que o Xamarin não possui um emulador para os sistemas iOS. Outra limitação do projeto está relacionada ao tempo disponível para esmiuçar todas as possibilidades oferecidas pelo framework Xamarin.Forms e dessa forma poder fazer uma avaliação melhor das funcionalidades disponibilizadas para os desenvolvedores. Talvez pelo fato do aplicativo criado ser bem simples tenha contribuído para não necessitar ir mais a fundo em algum aspecto do framework. Além dessas limitações pode ser citada também a limitação de usuários para realizar os testes finais.

4 AVALIAÇÃO DO APLICATIVO

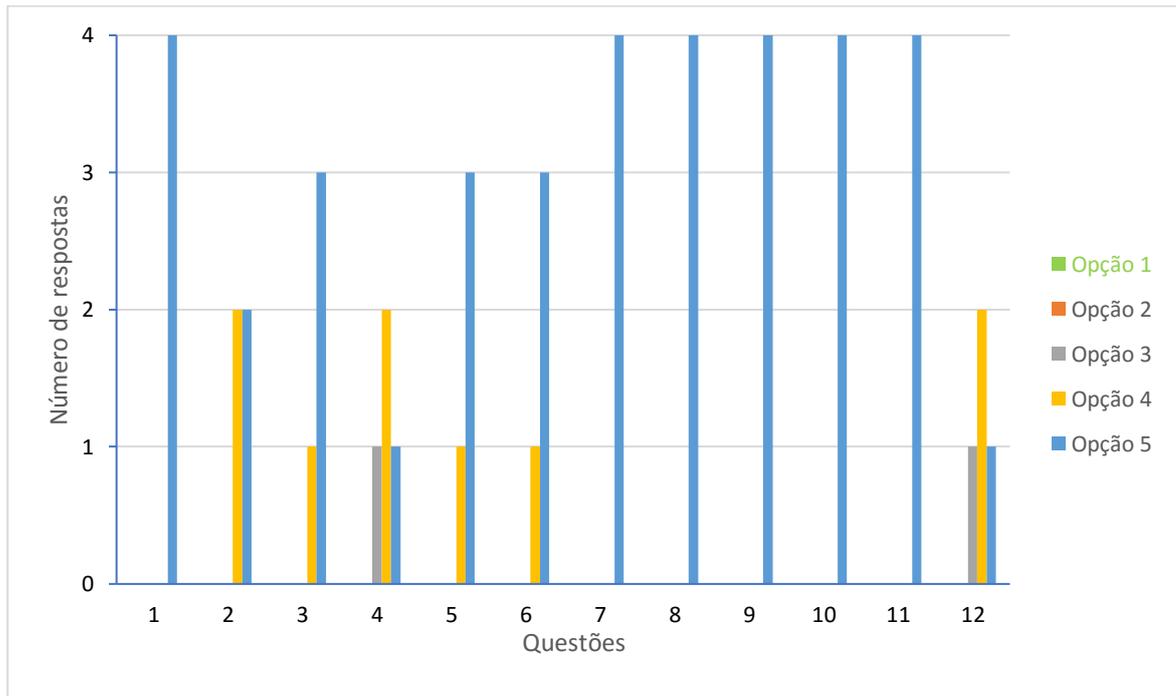
Para fazer uma avaliação e validar o aplicativo criado, uma pesquisa foi feita com os especialistas, as mesmas pessoas que auxiliaram na criação das estórias de usuário e protótipos das telas. Nesta avaliação foi desenvolvido um questionário que pode ser encontrado no Anexo A. Ao todo foram quatro usuários que participaram dos testes de avaliação que foram realizados por engenheiros agrônomos da Faculdade de Agronomia da UFRGS.

O cenário de teste consistiu em realizar as tarefas do questionário usando um dispositivo móvel, a opção de usar diretamente no dispositivo móvel se deu pelo fato que dessa forma o usuário do aplicativo consegue fazer uma avaliação real de usabilidade do aplicativo, melhorando o resultado da avaliação. Nesta configuração o aplicativo estava rodando no dispositivo móvel em modo debug, reduzindo assim um pouco o seu desempenho. Por uma questão de disponibilidade de diferentes dispositivos todos os usuários realizaram os testes na plataforma móvel Android usando o mesmo aparelho (Samsung Galaxy J7).

Cada um dos especialistas recebeu um laudo de análise de solo conforme já apresentado na Figura 12, os dados de uma cultura (Anexo C) e um questionário (Anexo A). De posse desses dados o teste era iniciado e consistia em uma etapa prática, onde as três atividades eram realizadas usando o aplicativo no dispositivo. Em um segundo momento uma comparação dos dados de saída do aplicativo com os dados calculados manualmente, dessa forma validando os resultados. E por fim, respondiam o questionário marcando para cada uma das questões a opção conveniente, e também havia um campo aberto para comentários e considerações finais.

A Figura 23 ilustra as respostas compiladas para cada questão feita aos especialistas sobre o aplicativo. As respostas variaram de **(1) Discordo totalmente** a **(5) Concordo totalmente**

Figura 23: Opiniões dos especialistas referentes ao uso do aplicativo desenvolvido



Fonte: Próprio autor

Conforme os resultados apresentados na Figura 19 pode-se dizer que o nível de aceitação do aplicativo foi bem elevado, na maior parte das questões a resposta mais marcada foi a opção 5.

Na facilidade de encontrar o caminho para executar as tarefas solicitadas (Questão 1) todos os avaliadores marcaram que Concordam Totalmente. Para o fácil entendimento do cadastro de culturas (Questão 2) a avaliação foi dividida, metade Concorda Totalmente e a outra metade Concorda com Reservas. A inserção dos dados foi completamente tranquila para a maior parte dos avaliadores (Questão 3). As mensagens de alerta apresentadas pelo aplicativo ajudaram os especialistas, porém, elas poderiam ajudar de maneira mais satisfatória (Questão 4). Os nomes dos campos estavam condizentes com as informações que eles recebem para a maioria dos avaliadores (Questão 5). A Questão 6 referente a entrada dos dados do laudo de análise foi de fácil entendimento para a maior parte dos especialistas que informaram que essa tarefa foi transparente. A avaliação da escolha dos componentes de interface gráfica facilitou o uso do aplicativo para todos os avaliadores (Questão 7). Os resultados gerados, quando comparados com os calculados manualmente, estão corretos para todos os especialistas (Questão 8). A Questão 9 que faz a avaliação da apresentação dos resultados finais foi classificada como satisfatória para todos os especialistas. Para todas as avaliações o aplicativo facilita o trabalho do engenheiro agrônomo no trabalho de interpretar os laudos de análise (Questão 10). No

questão recomendação (Questão 11) todos os especialistas indicariam o aplicativo para os colegas de profissão. O desempenho do aplicativo (Questão 12) foi classificado como satisfatório pela maioria dos avaliadores.

Por fim um campo aberto a comentários e recomendações adicionais foi disponibilizado aos avaliadores. Segue uma lista com as sugestões e críticas apresentadas:

- Na tela de entrada dos dados do laudo inserir unidades de medida nos rótulos dos campos. Essa solicitação foi sugerida pois se um laudo vier a ser gerado com unidades de medidas diferentes das usadas por padrão, pode gerar um erro na entrada dos dados se os mesmos não forem convertidos para a nova unidade.
- Expandir a avaliação dos índices de fertilidade e recomendação de adubação para os micronutrientes, por exemplo: Mg, Cu, B, Zn, S, Ca, Mn, Fe e Na. No aplicativo criado foram calculados os índices de fertilidade e recomendação de adubação para os macros nutrientes (N, P e K), ficando a cargo do engenheiro agrônomo calcular manualmente os valores para os demais nutrientes.
- Ter mais uma opção no cadastro de cultura para receber a expectativa de rendimento. Cultura do arroz traz essa informação. Para algumas culturas dependendo a expectativa de rendimento informada pelo produtor há um acréscimo na quantidade de fertilizante a ser aplicado para atingir essa meta de colheita.
- Depois da recomendação de adubação ter a possibilidade de adicionar as quantidades para manutenção e reposição baseada na expectativa de rendimento. Conforme o rendimento definido pelo agricultor além da indicação normal de aplicação também existe uma quantidade adicional de fertilizante que é dividida entre manutenção e reposição.
- Possibilitar o cadastro de período de aplicação dos nutrientes para cada fase da cultura, por exemplo: Pré-plantio, semeadura e cobertura. Além do cadastro das tabelas de adubação para a cultura, ser possível cadastrar a quantidade de fertilizante a ser aplicado em cada uma das fases de crescimento da cultura.

Todas as sugestões apresentadas pelos especialistas podem ser desenvolvidas em uma futura versão do aplicativo, é uma sugestão para um trabalho futuro, desta forma o aplicativo se tornaria mais completo atendendo de uma forma melhor as necessidades dos engenheiros agrônomos.

5 AVALIAÇÃO DO FRAMEWORK XAMARIN.FORMS

Neste capítulo serão discutidos os resultados da avaliação do framework Xamarin.Forms utilizado neste estudo. Com o uso do Xamarin.Forms em apenas dois meses foi possível criar uma aplicação móvel que roda nas três principais plataformas móveis do mercado atual (Android, iOS e Windows Phone), isso sem ter conhecimento prévio no desenvolvimento de aplicativos para dispositivos móveis, apenas conhecendo a linguagem C# e a IDE do Visual Studio. Isso foi muito interessante e demonstra como é grande o potencial do framework.

A comunidade Xamarin está crescendo bastante e é possível encontrar material para qualquer tipo de dúvida. A própria Xamarin possui um portal onde disponibiliza para os desenvolvedores todo o manual da plataforma com exemplos de como usar cada recurso, além de um fórum para discussão de dúvidas. A quantidade de bibliotecas e recursos disponíveis também está aumentando.

O uso de uma linguagem de programação única, C#, torna mais fácil e rápido o processo de desenvolvimento. Os recursos oferecidos pelo C#, como LINQ, Genéricos e Listas também proporcionam um ganho na produtividade por tornarem mais simples o tratamento dos dados do aplicativo. Mesmo com o uso de uma linguagem única de desenvolvimento os aplicativos criados usando o Xamarin.Forms são ditos nativos, pois conseguem explorar todo o potencial da plataforma em que estão rodando. Outra grande vantagem no uso do framework é o alto reuso de código, não apenas código base, mas também permitindo uma interface de usuário compartilhada entre todas as plataformas.

Ainda existem limitações no framework do Xamarin.Forms, a principal dela se dá no uso da plataforma iOS, pois sem ter um dispositivo da Apple não é possível rodar a aplicação e ver o resultado. Não existe um emulador para essa plataforma, assim como existem emuladores para Android e Windows Phone. Os emuladores facilitam muito o trabalho do desenvolvedor tornando possível os testes sem usar um dispositivo físico. Outra limitação é que a interface de usuário compartilhada oferecida pelo Xamarin.Forms ainda é limitada nos componentes e funcionalidades disponíveis, se a aplicação exigir uma interface mais complexa ainda é necessário implementar código específico.

O Xamarin.Forms surge como uma ótima alternativa para o desenvolvimento de aplicativos, por possibilitar que os sistemas criados possam rodar em multiplataforma com um grande reuso de código implementado. Apesar de suas atuais limitações, ele tende a ficar mais

completo com o passar do tempo. Isso se deve em grande parte pela aquisição da Xamarin pela Microsoft e pela aproximação da Microsoft com a Apple se espera que novas funcionalidades acabem surgindo e que a falta de recursos seja suprida.

6 CONCLUSÃO E TRABALHO FUTURO

O propósito deste trabalho foi fazer uma avaliação da eficiência, usabilidade, aplicabilidade e complexidade do framework Xamarin.Forms no desenvolvimento de aplicativos móveis multiplataforma, usando um código base único para as três principais plataformas móveis do mercado. Como objeto de estudo um aplicativo foi criado usando esta ferramenta. Este aplicativo tem por objetivo facilitar o trabalho dos engenheiros agrônomos na interpretação de laudos de análise de solo e gerar os índices de fertilidade e uma recomendação de adubação e calagem para a amostra de solo à qual o laudo se refere.

Durante o desenvolvimento do trabalho foi possível mapear alguns pontos fortes e algumas deficiências do framework Xamarin.Forms. Como pontos fortes podemos destacar a grande reutilização de código, dados obtidos indicam que esses valores podem chegar a 90%. Uso de linguagem única (C#) para desenvolvimento da aplicação, torna mais simples o trabalho do desenvolvedor que necessita conhecer apenas um ambiente de desenvolvimento. Oferecer a possibilidade de criação de uma interface de usuário comum para as três plataformas, além de melhorar desempenho do desenvolvedor evita tratamentos diferentes entre as plataformas. Outro ponto muito forte desse framework é a grande disponibilidade de material na internet, o próprio Xamarin disponibiliza um canal para os desenvolvedores contendo a documentação completa para consulta, com exemplos, e um fórum para os programadores postarem perguntas ou responder as dúvidas de outros desenvolvedores. Os pontos fracos são limitações para criação da interface, nem tudo pode ser feito com os componentes de interface disponibilizados pelo Xamarin.Forms. A falta de emulador para a plataforma iOS, essa é uma grande limitação imposta, visto que sem um dispositivo Apple não é possível ver o resultado final.

Mesmo com as deficiências citadas o uso do Xamarin.Forms facilita o processo de desenvolvimento de aplicativos móveis multiplataforma. Apenas com o conhecimento da linguagem C# foi possível construir uma aplicação completa reutilizando a maior parte do código para as três principais plataformas móveis do mercado. É um modelo de desenvolvimento que vai ganhar cada vez mais espaço no mercado, por ser simples e gerar ótimos resultados.

Como trabalho futuro podem ser citadas as melhorias descritas pelos especialistas, além de poder usar mais funcionalidades do framework Xamarin.Forms e com isso possibilitar uma melhor avaliação. Também vai ser importante conseguir testar o aplicativo na plataforma iOS.

REFERÊNCIAS

- BALSAMIQ MOCKUPS. Balsamiq Mockups, 2016. Disponível em <<https://balsamiq.com/products/mockups/>>. Acessado em 21 nov. 2016.
- CQFS-RS/SC (Comissão de Química e Física do Solo dos Estados do RS e SC). **Manual de adubação e calagem para os Estados do Rio Grande do Sul e de Santa Catarina**. 10ª ed. Porto Alegre: SBCS – Núcleo Regional Sul, 2004. 400p.
- DEVMEDIA. Entendendo o Pattern Model View ViewModel MVVM, 2014. Disponível em <<http://www.devmedia.com.br/entendendo-o-pattern-model-view-viewmodel-mvvm/18411>>. Acessado em 21 nov. 2016.
- FELIX R. SQLite - O pequeno Notável, 2016. Disponível em <<http://www.devmedia.com.br/sqlite-o-pequeno-notavel/7249>>. Acessado em 21 nov. 2016.
- MSDN MICROSOFT. Xamarin e Visual Studio, 2016 a. Disponível em <<https://msdn.microsoft.com/pt-br/library/Mt299001.aspx>>. Acessado em 21 nov. 2016.
- MSDN MICROSOFT. Introdução a Consultas LINQ (C#), 2016 b. Disponível em <<https://msdn.microsoft.com/pt-br/library/bb397906.aspx>>. Acessado em 21 nov. 2016.
- MSDN MICROSOFT. Genéricos (Guia de Programação C#), 2016 c. Disponível em <<https://msdn.microsoft.com/pt-br/library/512aeb7t.aspx>>. Acessado em 21 nov. 2016.
- MSDN MICROSOFT. Classe List<T>, 2016 d. Disponível em <[https://msdn.microsoft.com/pt-br/library/6sh2ey19\(v=vs.110\).aspx](https://msdn.microsoft.com/pt-br/library/6sh2ey19(v=vs.110).aspx)>. Acessado em 21 nov. 2016.
- MSDN MICROSOFT. The MVVM Pattern, 2016 e. Disponível em <<https://msdn.microsoft.com/en-us/library/hh848246.aspx>>. Acessado em 21 nov. 2016.
- MSDN MICROSOFT. Visual Studio IDE, 2016 f. Disponível em <<https://msdn.microsoft.com/pt-br/library/dn762121.aspx>>. Acessado em 21 nov. 2016.
- OPUS SOFTWARE. Desenvolvimento em Xamarin: Nossa Experiência, 2016. Disponível em <<http://www.opus-software.com.br/desenvolvimento-em-xamarin/>>. Acessado em 21 nov. 2016.
- RADI, A. A. Evaluation of Xamarin Forms for Multiplatform Mobile Application Development, **Technical Library Paper**, 2016. Disponível em

<<http://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1238&context=cistechlib>>. Acesso em 21 nov. 2016.

SQLITE. About SQLite. Disponível em <<https://sqlite.org/about.html>>. Acessado em 21 nov. 2016.

TFS MICROSOFT. Team Foundation Server. Disponível em <<https://www.visualstudio.com/tfs/>>. Acessado em 21 nov. 2016.

WENZEL, J. Burn Down Chart Tutorial: Simple Agile Task Tracking, **Joel in Point Form**, 2010. Disponível em <<http://joel.inpointform.net/software-development/burn-down-charts-tutorial-simple-agile-project-tracking/>>. Acesso em 21 nov. 2016.

XAMARIN DEVELOPER. Cross-Platform Performance, 2016 a. Disponível em <https://developer.xamarin.com/guides/cross-platform/deployment_testing_and_metrics/memory_perf_best_practices/>. Acesso em 21 nov. 2016.

XAMARIN DEVELOPER. Xamarin.Forms Controls Reference, 2016 b. Disponível em <<https://developer.xamarin.com/guides/xamarin-forms/controls/>>. Acesso em 21 nov. 2016.

XAMARIN DEVELOPER. Xamarin.Forms Pages, 2016 c. Disponível em <<https://developer.xamarin.com/guides/xamarin-forms/controls/pages/>>. Acesso em 21 nov. 2016.

XAMARIN DEVELOPER. Xamarin.Forms Layouts, 2016 d. Disponível em <<https://developer.xamarin.com/guides/xamarin-forms/controls/layouts/>>. Acesso em 21 nov. 2016.

ANEXO A

Abaixo se encontra o questionário aplicado aos especialistas na avaliação do aplicativo. Ele é composto por atividades que foram desenvolvidas pelos engenheiros agrônomos e as questões de múltipla escolha que deram origem aos dados apresentados no trabalho. Além do campo aberto a sugestões.

Questionário para avaliação do aplicativo

Para avaliação do aplicativo pelo engenheiro agrônomo foram definidas algumas tarefas simples para serem executadas. As tarefas estão definidas abaixo.

- Tarefa 1: Inserir uma cultura no cadastro. Definindo seus dados básicos e suas tabelas de Nitrogênio, Fósforo e Potássio. (Serão fornecidas as tabelas da cultura conforme Manual de Recomendação de Adubação e Calagem)
- Tarefa 2: Entrada dos dados de um laudo de análise de solo (será fornecido o laudo de análise de uma amostra de solo) e usando a cultura previamente cadastrada fazer a interpretação dos resultados: índices de fertilidade, calagem e adubação.
- Tarefa 3: Fazer a comparação dos resultados gerados pelo aplicativo com os dados calculados manualmente pelo engenheiro agrônomo.

Com base nas tarefas anteriores responder o questionário abaixo.

Para cada questão escolher uma opção variando entre **(1) Discordo totalmente** e **(5)**

Concordo totalmente

- 1) Na sua opinião, foi fácil encontrar o caminho para executar as tarefas solicitadas
1 () 2 () 3 () 4 () 5 ()
- 2) Na sua opinião, foi fácil o entendimento do cadastro de culturas
1 () 2 () 3 () 4 () 5 ()
- 3) A inserção dos dados no cadastro de culturas se deu de uma forma tranquila:
1 () 2 () 3 () 4 () 5 ()

- 4) As mensagens de alerta do aplicativo ajudaram a entender o que estava acontecendo
1 () 2 () 3 () 4 () 5 ()
- 5) Os nomes dos campos estão condizentes com as informações que eles recebem
1 () 2 () 3 () 4 () 5 ()
- 6) Na entrada dos dados do laudo de análise, foi fácil entender o que deve ser inserido em cada campo
1 () 2 () 3 () 4 () 5 ()
- 7) Os componentes escolhidos (caixas de entrada de texto, campos de escolha) facilitaram o uso do aplicativo
1 () 2 () 3 () 4 () 5 ()
- 8) Os resultados gerados estão corretos, quando comparados aos calculados manualmente pelo engenheiro agrônomo
1 () 2 () 3 () 4 () 5 ()
- 9) A forma como os resultados foram apresentados foi satisfatória
1 () 2 () 3 () 4 () 5 ()
- 10) Este aplicativo facilita o trabalho do Engenheiro agrônomo na interpretação dos laudos de análise
1 () 2 () 3 () 4 () 5 ()
- 11) Você recomendaria este aplicativo para seus colegas de profissão
1 () 2 () 3 () 4 () 5 ()
- 12) O desempenho do aplicativo foi satisfatório
1 () 2 () 3 () 4 () 5 ()

Campo aberto para comentários, recomendações e sugestões:

ANEXO B

Neste anexo podem ser encontradas as tabelas extraídas do Manual de Adubação e Calagem (Rolas, 2004). Essas tabelas serviram de base para a criação da regra de negócio do aplicativo.

Tabela 5.1. Interpretação de valores de pH em água, saturação da CTC por bases e por alumínio

Interpretação	pH em água	Saturação por bases (CTC _{pH 7,0})	Saturação por alumínio (CTC _{efetiva})
		----- % -----	
Muito baixo	≤ 5,0	< 45	< 1
Baixo	5,1 – 5,4	45 – 64	1 – 10
Médio	5,5 – 6,0	65 – 80	10,1 – 20
Alto	> 6,0	> 80	> 20

Os valores analíticos referem-se somente às faixas de interpretação especificadas, não havendo, necessariamente, correspondência entre si. Por exemplo: em valores altos de pH (> 6,0), geralmente a saturação por bases é alta (> 80 %), mas a saturação por alumínio é muito baixa (< 1 %)

Tabela 5.2. Interpretação de teores de argila e de matéria orgânica e da capacidade de troca de cátions (CTC) a pH 7,0

Argila		Matéria orgânica		CTC _{pH 7,0}	
Faixa	Classe	Faixa	Interpretação	Faixa	Interpretação
%		%		cmol _c /dm ³	
≤ 20	4	≤ 2,5	Baixo	≤ 5,0	Baixo
21 - 40	3	2,6 - 5,0	Médio	5,1 - 15,0	Médio
41 - 60	2	> 5,0	Alto	> 15,0	Alto
> 60	1	-	-	-	-

Tabela 5.5. Interpretação do teor de potássio conforme as classes de CTC do solo a pH 7,0

Interpretação	CTC _{pH 7,0} (cmol _c /dm ³)		
	> 15,0	5,1 - 15,0	< 5,0
----- mg de K/dm ³ -----			
Muito baixo	≤ 30	≤ 20	≤ 15
Baixo	31 - 60	21 - 40	16 - 30
Médio	61 - 90	41 - 60	31 - 45
Alto	91 - 180	61 - 120	46 - 90
Muito alto	> 180	> 120	> 90

Tabela 6.1. Classificação de espécies em relação ao pH do solo

pH de referência ⁽¹⁾	Culturas
pH 6,5	Alfafa, aspargo, piretro.
pH 6,0	Abacateiro, abóbora, alcachofra, alface, alho, almeirão ameixeira, amendoim, arroz de sequeiro, aveia, bananeira, batata-doce, beterraba, brócolo, cana-de-açúcar, camomila, canola, caqui, cebola, cenoura, cevada, chicória, citros, consorciação de gramíneas e leguminosas de estação fria, couve-flor, crisântemo de corte, ervilha, estévia, feijão, figueira, fumo, girassol, hortelã, leguminosas forrageiras de estação fria, leguminosas forrageiras de estação quente, consorciação de gramíneas e leguminosas de estação quente, linho, macieira, maracujazeiro, melancia, melão, milho, moranga, morangueiro, nectarineira, nogueira-pecã, painço, pepino, pereira, pessegueiro, pimentão, quivizeiro, rabanete, repolho, roseira de corte, rúcula, soja, sorgo, tomate, tremoço, trigo, triticale, urucum, vetiver, videira.
pH 5,5	Abacaxizeiro, acácia negra, alfavaca, amoreira-preta, arroz irrigado no sistema de semeadura em solo seco, batata, bracatinga, calêndula, camomila, capim elefante, cardamomo, carqueja, coentro, curcuma, erva-doce, eucalipto, funcho, gramíneas forrageiras de estação fria, gramíneas forrageiras de estação quente, gengibre, manjerição, pinus, salsa.
— ⁽²⁾	Capim-limão, citronela-de-Java, palma-rosa e chá
Sem correção da acidez ⁽³⁾	Arroz irrigado no sistema pré-germinado ou com transplante de mudas, erva-mate, mandioca, mirtilo, pastagem natural, araucária.

⁽¹⁾ Em geral, no sistema plantio direto, a maioria das culturas de grãos desenvolve-se adequadamente em solos com pH 5,5, desde que a saturação da CTC por bases seja maior do que 65%.

⁽²⁾ A calagem é indicada quando a saturação da CTC por bases for menor do que 50%.

⁽³⁾ Aplicar 1 t/ha de calcário quando os teores de cálcio ou de magnésio forem inferiores aos da classe "Médio" (Tabela 5.6), exceto para o mirtilo para o qual não se recomenda calagem.

Tabela 6.2. Quantidades de calcário necessárias para elevar o pH em água do solo a 5,5, 6,0 e 6,5, estimadas pelo índice SMP⁽¹⁾

Índice SMP	pH desejado		
	5,5	6,0	6,5
	----- t/ha ⁽²⁾ -----		
≤ 4,4	15,0	21,0	29,0
4,5	12,5	17,3	24,0
4,6	10,9	15,1	20,0
4,7	9,6	13,3	17,5
4,8	8,5	11,9	15,7
4,9	7,7	10,7	14,2
5,0	6,6	9,9	13,3
5,1	6,0	9,1	12,3
5,2	5,3	8,3	11,3
5,3	4,8	7,5	10,4
5,4	4,2	6,8	9,5
5,5	3,7	6,1	8,6
5,6	3,2	5,4	7,8
5,7	2,8	4,8	7,0
5,8	2,3	4,2	6,3
5,9	2,0	3,7	5,6
6,0	1,6	3,2	4,9
6,1	1,3	2,7	4,3
6,2	1,0	2,2	3,7
6,3	0,8	1,8	3,1
6,4	0,6	1,4	2,6
6,5	0,4	1,1	2,1
6,6	0,2	0,8	1,6
6,7	0	0,5	1,2
6,8	0	0,3	0,8
6,9	0	0,2	0,5
7,0	0	0	0,2
7,1	0	0	0

⁽¹⁾ Análise conjunta baseada nos trabalhos de Murdock et al. (1969); Kaminski (1974); Scherer (1976); Ernani & Almeida (1986); Anjos et al. (1987) e Ciprandi et al. (1994).

⁽²⁾ Calcário com PRNT 100%.

ANEXO C

A seguir se encontram os dados da cultura do Trigo, cada cultura possui estas informações que possibilitam o cálculo da recomendação de adubação.

10.21 - TRIGO

Calagem

No sistema plantio direto, utilizar as indicações de calagem constantes na página 115 e na Tabela 6.3. No sistema convencional, adicionar a quantidade de calcário indicada pelo índice SMP para o solo atingir pH 6,0 (Tabela 6.2).

Nitrogênio

Teor de matéria orgânica no solo	Nitrogênio	
	Cultura antecedente	
	Leguminosa	Gramínea
%	----- kg de N/ha -----	
≤ 2,5	60	80
2,6 - 5,0	40	60
> 5,0	≤ 20	≤ 20

Para a expectativa de rendimento maior do que 2 t/ha, acrescentar aos valores da tabela 20 kg de N/ha em trigo após leguminosa e 30 kg de N/ha em trigo após gramínea, por tonelada de grãos a serem produzidos.

Aplicar entre 15 e 20 kg de N/ha na semeadura e o restante em cobertura, entre os estádios de afilamento e de alongamento (aproximadamente entre 30 e 45 dias após a emergência). Para as doses mais elevadas, pode-se parcelar a cobertura em duas aplicações, sendo uma no início do afilamento e a outra no início do alongamento. No caso de resteva de milho e, quando há muita palha, convém antecipar a adubação nitrogenada de cobertura, especialmente em solos arenosos ou com baixos teores de matéria orgânica.

Para cultivares muito suscetíveis ao acamamento devem ser utilizadas doses menores que as indicadas acima. Nas regiões de clima mais quente (região das Missões do Estado do RS, por exemplo), de menor altitude, e quando o trigo for antecedido pela soja, é recomendável restringir a aplicação de N a no máximo 40 kg/ha (base + cobertura) independentemente do teor de matéria orgânica do solo, a fim de evitar danos por acamamento. Nas regiões mais frias e solos com alto teor de matéria orgânica (Campos de Cima da Serra do RS), as doses de N indicadas podem ser aumentadas visando a expressão do potencial de rendimento.

Quando for cultivado nabo forrageiro como cultura intercalar entre o milho e o trigo e a fitomassa produzida pelo nabo for maior do que 3 t/ha, sugere-se aplicar a dose de N indicada para o trigo cultivado após leguminosa. Esta recomendação não é válida quando o nabo também for cultivado antes do milho.

Fósforo e potássio⁽¹⁾

Interpretação do teor de P ou de K no solo	Fósforo por cultivo		Potássio por cultivo	
	1º	2º	1º	2º
	kg de P ₂ O ₅ /ha		kg de K ₂ O/ha	
Muito baixo	110	70	100	60
Baixo	70	50	60	40
Médio	60	30	50	20
Alto	30	30	20	20
Muito alto	0	≤ 30	0	≤ 20

Para a expectativa de rendimento maior do que 2 t/ha, acrescentar aos valores da tabela 15 kg de P₂O₅/ha e 10 kg de K₂O/ha, por tonelada adicional de grãos a serem produzidos.

⁽¹⁾ Ver os itens 7.4 e 7.9.