# Material e métodos

A seção material e métodos foi dividida entre os material e métodos relativos ao software desenvolvido e o material e métodos relativos ao estudo de caso, que é a aplicação do software desenvolvido.

## Software

O Software proposto foi denominado Gafanhoto e sua verão atual de desenvolvimento é a 1.0.0. As tecnologias e metodologias utilizadas para o desenvolvimento deste software estão descritas nas subseções a seguir.

### Requisitos de software

O Gafanhoto V:1.0.0 possui apenas uma distribuição e essa distribuição foi compilada para rodar no sistema operacional Windows 7 x64. Dado continuidade ao desenvolvimento deste software, as versões serão compiladas também para ambientes Linux x64.

As especificações mínimas de hardware variam muito dependendo da área de estudo e das resoluções espaciais e temporais utilizadas. As especificações mínimas recomendadas para o estudo de caso descrito neste trabalho possibilitando a conclusão de todos os procedimentos em menos de um dia (24 horas) são:

* Processador: 4 núcleos à 2.2Ghz;
* Armazenamento: 50GB;
* Velocidade de armazenamento (leitura e gravação) 200MB/s.

### Python

A linguagem de programação utilizada para construir o software aqui apresentado foi a linguagem Python V:2.7.9, por fornecer maior compatibilidade com diversas distribuições de bibliotecas interessantes ao escopo do software. A linguagem está disponível para download na URL: https://www.python.org/downloads/release/python-279/. Sua instalação não é desnecessária para a execução do mesmo.

### GDAL

A versão da biblioteca de tradução e abstração de rasters e vetores GDAL utilizada, para o Gafanhoto V:1.0.0, é a distribuição gdal-201-1800-x64-core.msi, que está disponível para download na URL: http://www.gisinternals.com/query.html?content=filelist&file=release-1800-x64-gdal-mapserver.zip. Esta distrubuição consiste em um instalador genérico para os componentes do núcleo (core) da GDAL. Este core é necessário para total o funcionamento total do software Gafanhoto V:1.0.0, ele fornece diversos recursos de linha de comando que são utilizados no software como interpolação e outros. As informações sobre a instalação e configuração entontaram-se em Anexo.

### Qt Designer

Qt Designer é uma das ferramentas fornecidas pela Qt. Utilizada para construção de UI, ela fornece um código único descritivo de interface. A versão utilizada é a 4.5.1. Ela é utilizada para desenhar as interfaces de usuário (QT, 2015). O Qt designer é livre para aplicações não comercializadas. Sua instalação não é desnecessária para a execução do mesmo.

### Eclipse

Eclipse é uma IDE (IntegratedDevelopmentEnvironment), ela fornece um ambiente para programação e diversos recursos que serão aplicados para a otimização do processo de implementação e teste do software (Eclipse, 2015). A versão utilizada foi a Eclipse Luna. Sua instalação não é desnecessária para a execução do mesmo.

### Bibliotecas diversas

As bibliotecas utilizadas na construção do software Gafanhoto V1.0.0, sãs as seguintes:

* Pip: utilizada para fazer download e atualização de outras bibliotecas;
* GDAL-1.11.2-cp27-none-win\_amd64: tradução e abstração de rasters e layers;
* lxml-3.4.2-cp27-none-win\_amd64: utilizada para configurações de arquivos XML entre o core GDAL e o Gafanhoto;
* numpy-1.9.2rc1+mkl-cp27-none-win\_amd64: biblioteca de cálculos matemáticos;
* py2exe-0.6.10a1-cp27-none-win\_amd64: gera o arquivo executável para o programa;
* PyQt4-4.11.3-cp27-none-win\_amd64: desenho e formatação de User Initerfaces (UI), fornece uma série de recursos e componentes visuais para elaboração das telas de software para interação com o usuário;
* Fiona-1.5.0+gdal111-cp27-none-win\_amd64: para leitura edição e gravação de arquivos shapes(vetores) georreferenciados;
* rasterio-0.17.1+gdal111-cp27-none-win\_amd64: para leitura edição e gravação de arquivos rasters (matrizes) georreferenciados.

### Modelo Visão e Controle

O padrão MVC (Model-View-Controller) sugere dividir a arquitetura de software em componentes, tornando o desenvolvimento mais claro e enxuto e possibilitando, posteriormente, a reciclagem e manutenção do sistema com maior facilidade e segurança. Para que isso ocorra, deve haver independência dos componentes e ela só será atingida se houver uma organização do sistema em camadas para garantir a escalabilidade, eficiência e a reusabilidade (GAMMA et al. 2000; BALTHAZAR et al., 2007).

Para a organização e estrutura do software, foi utilizada arquitetura de desenvolvimento Modelo Visão Controle (MVC):

* Modelo: camada onde encontram-se a implementação das funções objetivo do software, a implementação das regras de negócio bem como todas as ferramentas internas de acesso a dados e manipulação dos dados.
* Visão: camada onde entontaram-se as implementações referentes as interfaces gráficas, ou interface de usuário, presentes no software. É nessa cama que as interações entre software e usuário acontecem.
* Controle: camada onde encontram-se as implementações relativas ao controle de fluxo de informação interna no software, é nesta camada que as informações são validadas, organizadas e então submetidas as funções.

### Padrões de projeto

Os padrões de projeto utilizados foram o Builder Method e Template Method.

Como um dos padrões comportamentais o *Template Method* auxilia na definição de um algoritmo com partes do mesmo definidos por métodos abstratos. As subclasses devem se responsabilizar por estas partes abstratas, deste algoritmo, que serão implementadas, possivelmente de várias formas, ou seja, cada subclasse irá implementar à sua necessidade e oferecer um comportamento concreto construindo todo o algoritmo. Maiores informações sobre estes e outros padrões de projeto podem ser encontradas em Gamma et al., (1995).

### Análise orientada a objetos usandoUML

A análise Orientada a Objetos (OO), com diagramas em UML será usada ao decorrer de todo o desenvolvimento do software. Ela será usada como ferramenta de elaboração, análise, e documentação do software. Permitindo que este se torne padronizado e de fácil entendimento, já que se destina não só a esta como também futuras pesquisas e implementações de modelos para a estimativa de produtividade agrícola.

O UML 2.2, conforme a OMG (2011), possui 14 tipos de diagramas, divididos em duas grandes categorias: Estruturais e Comportamentais. Sete tipos de diagramas representam informações estruturais, e os outros sete representam tipos gerais de comportamento, incluindo quatro diagramas em uma sub-categoria que representam diferentes aspectos de interação (Figura 1).

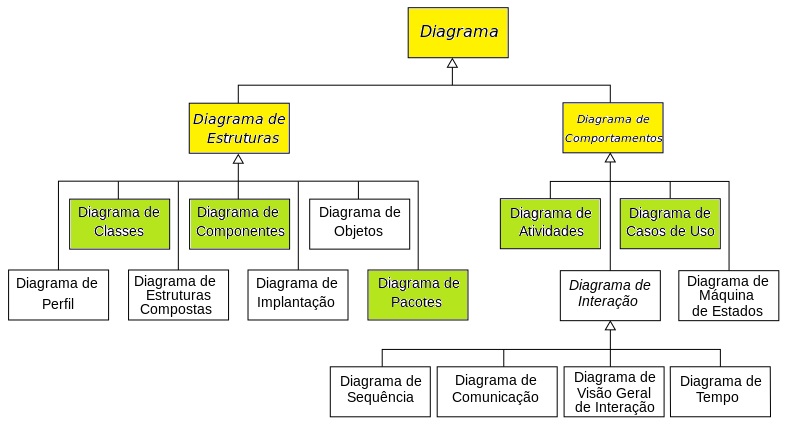


Figura 1 Diagramas UML e diagramas selecionados (adaptado de OMG, 2011)

A análise UML é extensa, sua complexidade pode tornar sua utilização completa em um empecilho quando se trata metodologias ágeis e softwares de estrutura relativamente simples, como o proposto. Portanto foram selecionados diagramas de maior utilidade, dado o contexto atual de desenvolvimento.

Os diagramas selecionados para o estudo (destacados em verde na Figura 1) foram:

* Diagrama de casos de uso: descreve as funcionalidades propostas para o novo sistema.
* Diagrama de pacotes: descreve os pacotes ou pedaços do sistema divididos em agrupamentos lógicos mostrando as dependências entre eles.
* Diagrama de componentes: ilustra como as classes deverão se encontrar organizadas através da noção de componentes de trabalho.
* Diagrama de classes: é uma representação da estrutura e relações entre classes.
* Diagrama de atividades: mostra o fluxo de controle de uma atividade para outra.

### Estrutura de referência dos dados

Para que o sistema tenha um nível de abstração que permita implementar diversos tipos de modelos, seja para balanço hídrico, como o que será aplicado ou em modelos de previsão de safra completos, todas as entradas de dados do sistema são abstraídas e divididas em categorias, as categorias abstraídas são mostradas na seção resultados.

### Padronização de ferramentas/operações

Todos os requisitos funcionais realizados pelo sistema, como interpolação e cálculos de BH, são abstraídos em uma estrutura padronizada de operações, de forma que possa ser implementado, sempre, novos recursos ao sistema, reaproveitando as funções existentes, é isso utilizado o padrão de projeto *Template* *Method*, padronizando as funções de forma que o sistema se torne modular.

## Estudo de caso

As duas principais *commodities* agrícolas do Brasil e do estado do Paraná são: a soja e o milho. O Brasil é o segundo maior produtor mundial de soja, atrás dos EUA. A produção brasileira de soja, no ano agrícola 2012/13, foi de 81,28 milhões de toneladas, superando em 22,4% a safra anterior, de 2011/12. O Paraná foi responsável por 19,5% da produção brasileira numa área plantada de 27,72 milhões de hectares. A produtividade média da soja brasileira foi de 2,93 t ha-1 superior 10% à da safra anterior. O Paraná teve a segunda maior produtividade brasileira de soja (3,34 t ha-1), numa área de 4,75 milhões de hectares (CONAB, 2013).

Para testar e validar o software criado ele foi utilizado para realizar a estimativa de produtividade da cultura da soja, devido à sua grande representação no Estado do Paraná, compreendendo os 399 municípios, localizados entre as Longitudes 48°2'W e 54°38'W e Latitudes 22°29'S e 26°43'S (Figura 2).

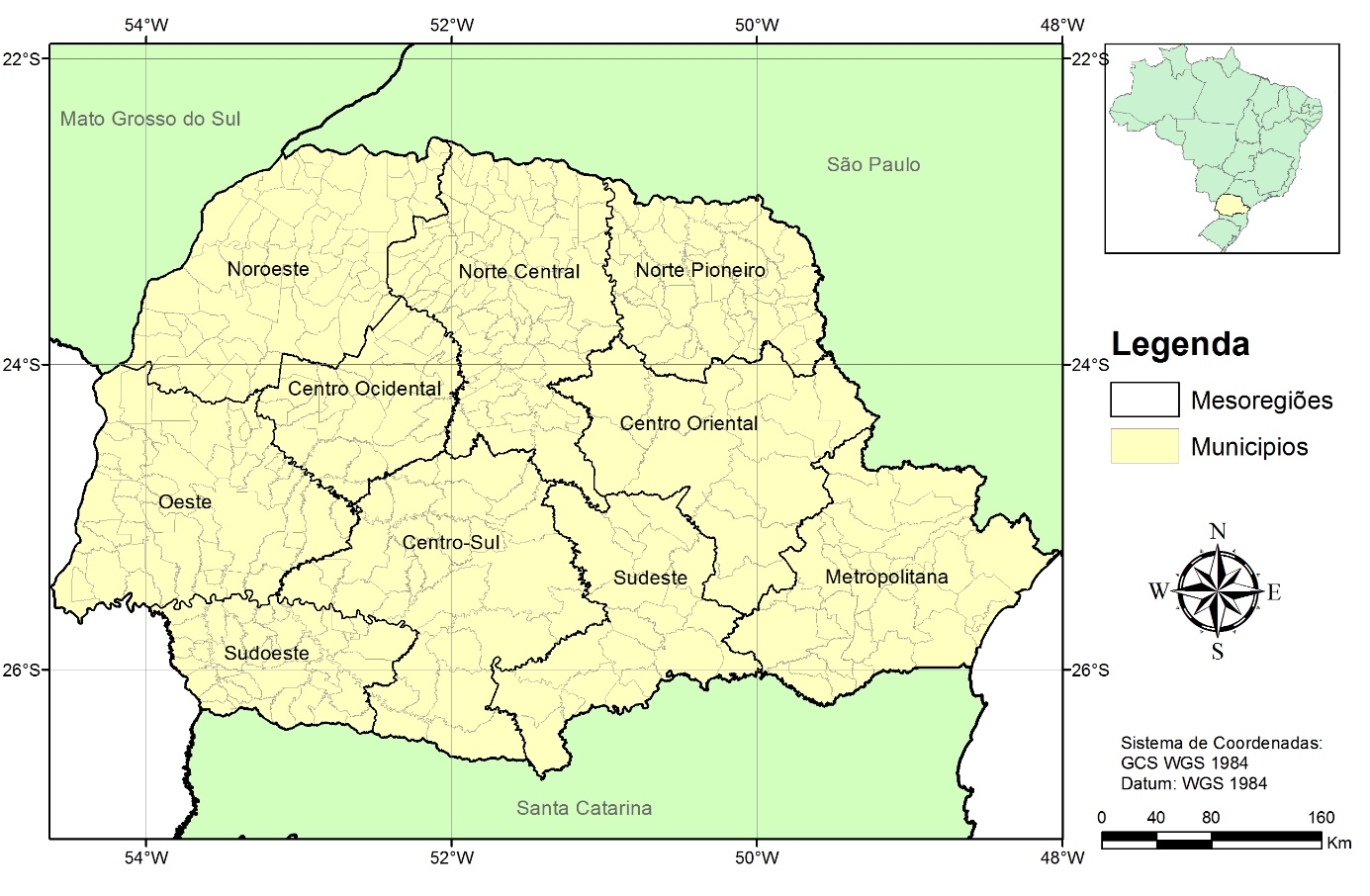


Figura 3 Mapa de localização dos 399 municípios e das 10 mesorregiões do estado do Paraná.

Para mapeamento dessa cultura, será utilizada a máscara de soja do ano-safra 2011-2012 feita por Souza (2013), ela foi obtida por sensoriamento remoto através de índices de NDVI provindos sensor MODIS (Figura 3).

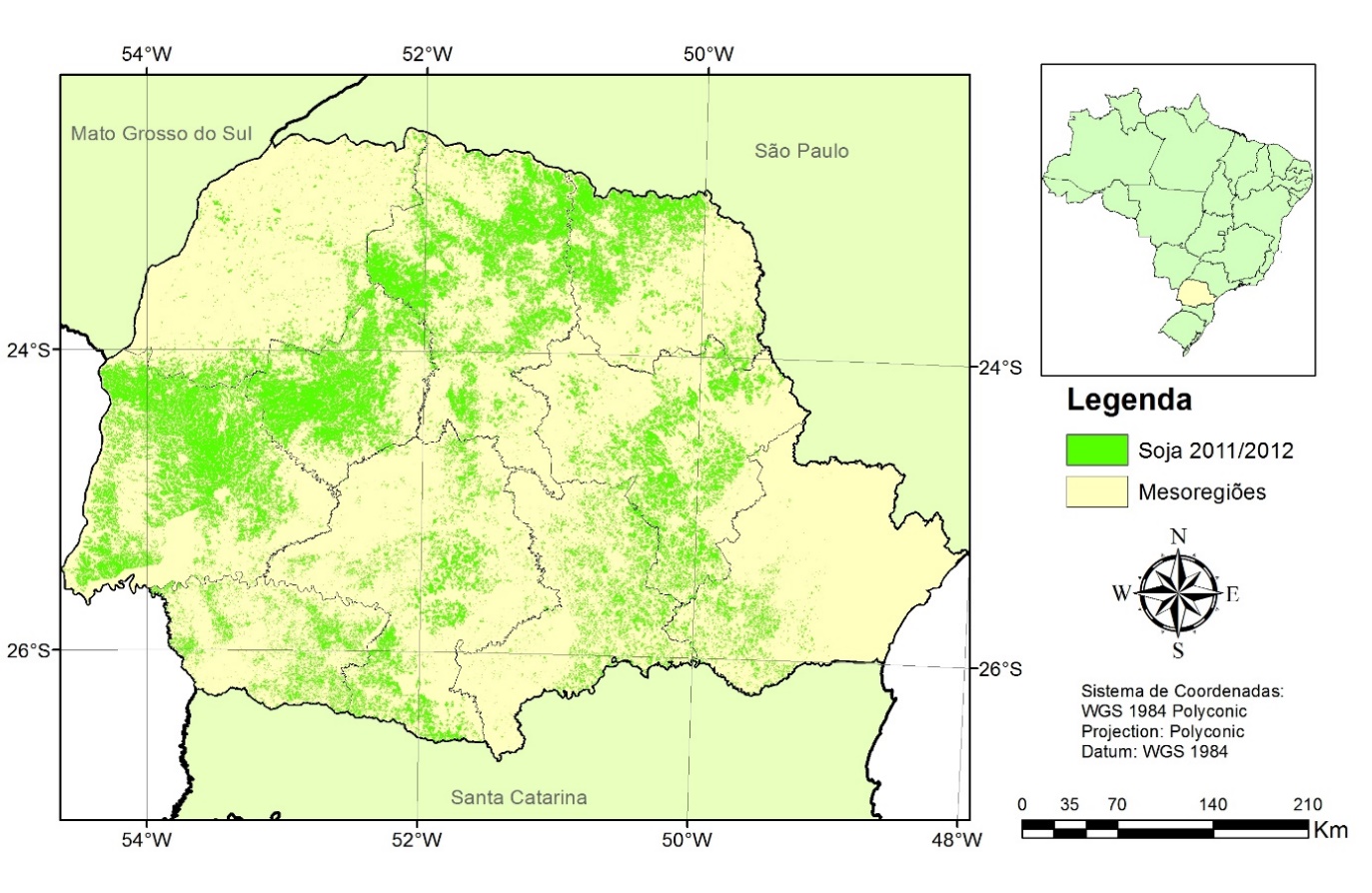


Figura 4 Área de soja para o ano safra 2011/2012 (adaptado de SOUZA et al., 2015).

Para a identificação das datas da cultura (colheita e semeadura) será utilizado o perfil do índice de EVI proveniente do sensor Modis, abordo dos satélites Terra e Aqua (resolução temporal de 8 dias e espacial de 250m). Para suavizar os ruídos serão utilizados os filtros Flat Smoother Series e Savitzky-Golay. O filtro Flat Smoother Series foi utilizado por Paloschi et al. (2015) suavizando o perfil espectro-temporal do índice de vegetação sem perder informações significativas.

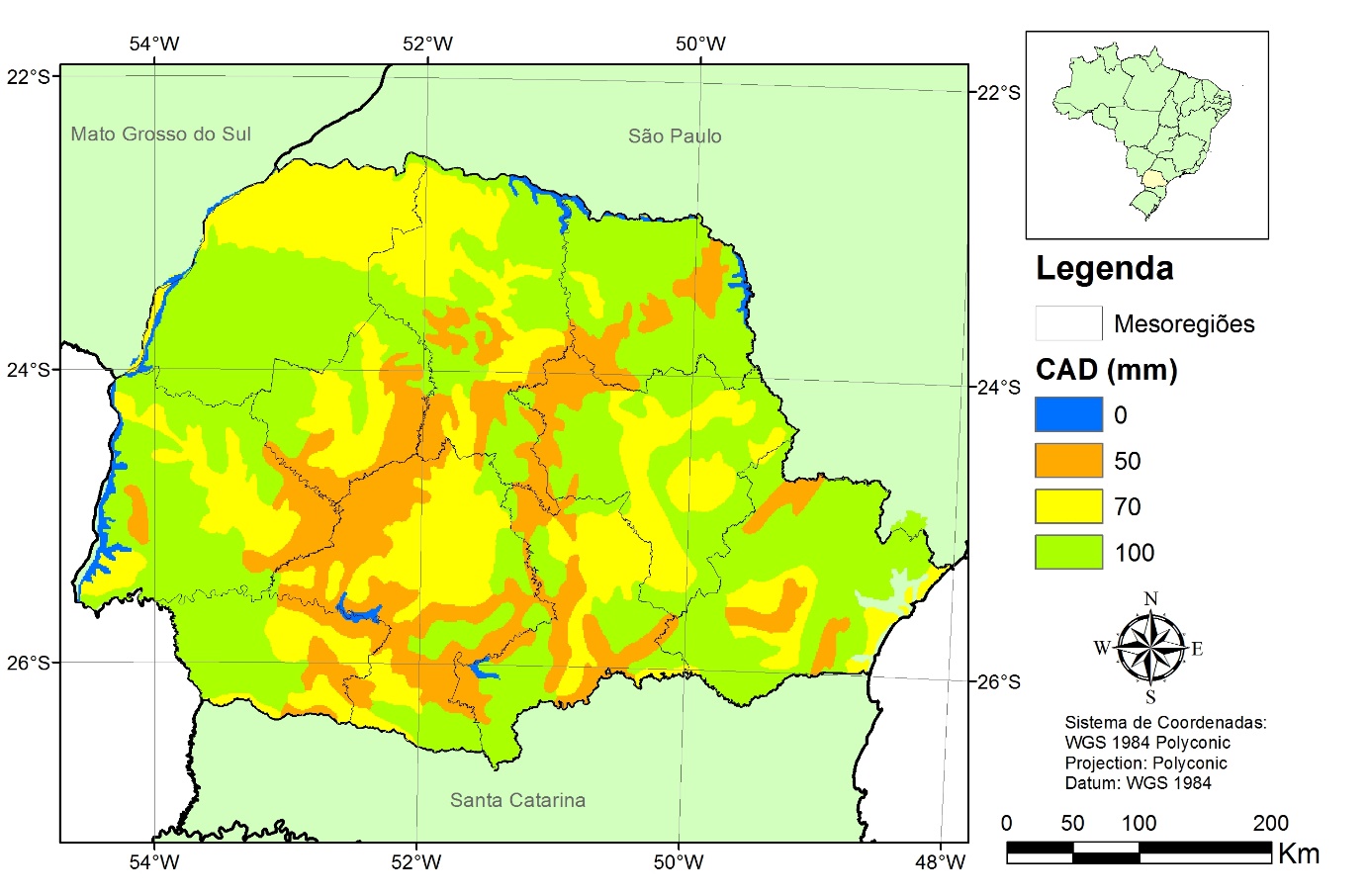
Becker et al. (2016) utilizou realizou a estimativa das datas relativas a cultura de soja, comparando com dados reais informados produtores do ano safra 2011/2012 e encontrou um ajuste de 3% para a semeadura, ou seja, um adiantamento de 4% do ponto de inflexão e um ajuste de -4% para a colheita, ou seja, um atraso de -3% em relação ao ponto de inflexão. Estes ajustes encontrados por Becker et al. (2016) serão utilizados para encontrar as datas de semeadura e colheita em toda a máscara.

A CAD é o intervalo de umidade do solo entre a capacidade de campo (CC%) e o ponto de murcha permanente (PMP%), tais dados dependem do tipo de solo de cada região (Richetti, 2014). Assim, serão utilizadas informações da capacidade de armazenamento de água do solo (CAD) geradas, conforme Farias et al. (2000) (Tabela 1), a partir dos tipos de solo obtidos do Novo Mapa de Solos do Brasil (EMBRAPA, 2011). Determinando assim, o valor de CAD para o estado do Paraná (Figura 9).

**Tabela 1 Capacidade de retenção de água no solo (CAD) para diferentes tipos de solo**

|  |  |
| --- | --- |
| Tipo de Solo | CAD (mm) |
| Neossolo Quartzarênico  Neossolo Flúvico de textura arenosa. | 50 |
| Latossolo Vermelho-Amarelo  Latossolo Vermelho (menos de 35% de argila)  Neossolo Litólico | 70 |
| Luvissolos  Argissolos  Nitossolos  Latossolos (exceto Latossolo Vermelho com menos de 35% de argila)  Cambissolos  Neossolo Flúvico de textura média a argilosa. | 100 |

FONTE: Adaptada de Embrapa (1999); Farias et al. (2000)



**Figura 9 Valores de CAD (mm) para o estado do Paraná. (Richetti, 2015)**

Os dados da cultura da soja, como profundidade do sistema radicular e coeficientes da cultura (*kc, ky, p*) serão obtidos do boletim nº 56 da FAO (ALLEN et al., 1998).

A cultura da soja foi considerada como padrão conforme descrita e caracterizada por Allen et al. (1998) (Tabela 2). Os coeficientes da cultura (*kc*) são apresentados para o estádio inicial, médio e final. Já o coeficiente de produtividade (*ky*), de profundidade máxima das raízes (Zr, em metros) e o fator de esgotamento *p* são fixos para todo ciclo da cultura. Estes valores serão fornecidos ao sistema da mesma forma, tabelados, apresentada.

**Tabela 2 Valores de referência para cultura da soja milho e trigo.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Culturas | Kc inicial | Kc médio | Kc final | Zr max | Ky | p |
| Soja | 0,40 | 1,15 | 0,50 | 0,60 | 0,80 | 0,50 |
| Milho | 0,40 | 1,20 | 0,60 | 1,00 | 0,80 | 0,55 |
| Trigo (Inverno) | 0,70 | 1,15 | 0,25 | 1,50 | 1,05 | 55 |

Fonte: Adaptado de ALLEN et al.(1998), Richetti (2015).

Os valores de kc e Zr (Tabela 3) utilizados foram atribuídos de acordo com o estádio fenológico de desenvolvimento da cultura, conforme metodologia apresentada por Johann (2012). Desta forma, considera-se que a cultura possui características diversas em suas diferentes fases, ou seja, a cultura possui maiores necessidades em seu florescimento do que em seu estabelecimento. Estes valores ponderados para os cultivares milho e trigo ainda não foram levantadas más serão incorporados.

**Tabela 3 Valores de referência ponderados por estádio fenológico das culturas de soja e milho.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Estádio de desenvolvimento da soja | | | | |
| Cultura | Estabelecimento | Vegetativo | Florescimento | Enchimento de Grãos | Colheita |
|  | Kc | | | | |
| Soja | 0,40 | 0,80 | 1,15 | 0,80 | 0,50 |
|  | Zr | | | | |
| Soja | 0,10 | 0,25 | 0,60 | 0,60 | 0,60 |

Fonte: Adaptado de Allen et al. (1998).