



BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

**SISTEMA WEB PARA AUXÍLIO NO PLANEJAMENTO DE
ROTAÇÕES DE CULTURAS**

WILLIAN PEREIRA DE SOUSA

Rio Verde, GO

2026



INSTITUTO FEDERAL GOIANO - CAMPUS RIO VERDE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

**SISTEMA WEB PARA AUXÍLIO NO PLANEJAMENTO DE
ROTAÇÕES DE CULTURAS**

WILLIAN PEREIRA DE SOUSA

Trabalho de Conclusão de Curso apresentado ao Instituto Federal Goiano - Campus Rio Verde, como requisito parcial para a obtenção do Grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. André da Cunha Ribeiro

Rio Verde, GO

Junho, 2026

**Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema Integrado de Bibliotecas do IF Goiano - SIBi**

S725 Sousa, Willian
Sistema web para auxílio no planejamento de rotações de
culturas / Willian Sousa. Rio Verde 2026.
40f. il.
Orientador: Prof. Dr. André da Cunha Ribeiro.
Tcc (Bacharel) - Instituto Federal Goiano, curso de 0219201 -
Bacharelado em Ciência da Computação - Integral - Rio Verde
(Campus Rio Verde).
1. Rotação de culturas. 2. Programação linear. 3. Target
MOTAD. 4. Sistema web. I. Título.

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO

PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS

NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

- | | |
|--|---|
| <input type="checkbox"/> Tese (doutorado) | <input type="checkbox"/> Artigo científico |
| <input type="checkbox"/> Dissertação (mestrado) | <input type="checkbox"/> Capítulo de livro |
| <input type="checkbox"/> Monografia (especialização) | <input type="checkbox"/> Livro |
| <input checked="" type="checkbox"/> TCC (graduação) | <input type="checkbox"/> Trabalho apresentado em evento |

Produto técnico e educacional - Tipo:

Nome completo do autor:

Willian Pereira de Sousa

Matrícula:

2017102201910102

Título do trabalho:

Sistema web para auxílio no planejamento de rotações de culturas

RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial: Não Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIF Goiano: **31 /07 /2026**

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro? Sim Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

- Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Documento assinado digitalmente
gov.br WILLIAN PEREIRA DE SOUSA
Data: 30/06/2026 11:57:35-0300
Verifique em <https://validar.iti.gov.br>

Rio Verde - GO
Local

30 /06 /2026
Data

Assinatura do autor e/ou detentor dos direitos autorais

Ciente e de acordo:

Assinatura do(a) orientador(a)

Documento assinado digitalmente

gov.br ANDRE DA CUNHA RIBEIRO
Data: 30/06/2026 12:07:10-0300
Verifique em <https://validar.iti.gov.br>

Regulamento de Trabalho de Curso (TC) – IF Goiano - Campus Rio Verde

ANEXO V - ATA DE DEFESA DE TRABALHO DE CURSO

Aos 25 dias do mês de junho de **dois mil e vinte e seis**, às 14 horas, reuniu-se a Banca Examinadora composta por: Prof. **André da Cunha Ribeiro** (orientador), Prof. **Marcio Antonio Ferreira Belo Filho** e Prof. **Márcio da Silva Vilela**, para examinar o Trabalho de Curso (TC) intitulado **SISTEMA WEB PARA AUXÍLIO NO PLANEJAMENTO DE ROTAÇÕES DE CULTURAS**, de **Willian Pereira de Sousa**, estudante do curso de **Bacharelado em Ciência da Computação** do IF Goiano – Campus Rio Verde, sob Matrícula nº **2017102201910102**. A palavra foi concedida à estudante para a apresentação oral do TC, em seguida houve arguição do candidato pelos membros da Banca Examinadora. Após tal etapa, a Banca Examinadora decidiu pela **APROVAÇÃO** do estudante. Ao final da sessão pública de defesa foi lavrada a presente ata, que, foi assinada pelos membros da Banca Examinadora e Mediador de TC.

Rio Verde, 25 de junho de 2026.

André da Cunha Ribeiro

Orientador

Marcio Antonio Ferreira Belo Filho

Membro da Banca Examinadora

Márcio da Silva Vilela

Membro da Banca Examinadora

Douglas Cedrim Oliveira

Mediador de TC

Documento assinado eletronicamente por:

- **Andre da Cunha Ribeiro, PROFESSOR ENS BASICO TECN TECNOLOGICO** , em 25/06/2026 15:43:50.
- **Marcio da Silva Vilela, PROFESSOR ENS BASICO TECN TECNOLOGICO** , em 25/06/2026 15:47:33.
- **Marcio Antonio Ferreira Belo Filho, PROFESSOR ENS BASICO TECN TECNOLOGICO** , em 25/06/2026 15:51:44.

Este documento foi emitido pelo SUAP em 25/06/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 834862

Código de Autenticação: ccc4cb8272



AGRADECIMENTOS

Primeiramente, agradeço a Deus pela saúde e perseverança que foram necessárias em todo o processo nesta jornada acadêmica.

A minha esposa, por todo apoio, incentivo e confiança que foram essenciais para continuar e finalizar este projeto.

Ao meu orientador, Prof. Dr. André da Cunha Ribeiro, pela orientação, paciência e principalmente pela perseverança para que este trabalho fosse concluído.

Ao Prof. Dr. Márcio Antônio Ferreira Belo Filho pela ajuda na investigação do artigo principal, sem este direcionamento não seria possível realizar este trabalho.

A todos que, de alguma forma, contribuíram para a construção e finalização deste trabalho, deixo minha sincera gratidão.

RESUMO

SOUSA, Willian. **Sistema web para auxílio no planejamento de rotações de culturas.** Junho, 2026. 40 f. Trabalho de Conclusão de Curso – (Curso de Bacharel em Ciência da Computação), Instituto Federal Goiano - Campus Rio Verde. Rio Verde, GO.

A rotação de culturas é uma das práticas agronômicas mais relevantes para a sustentabilidade dos sistemas de produção, mas o seu planejamento eficiente envolve múltiplas variáveis interdependentes — efeitos agronômicos entre culturas antecessoras e sucessoras, restrições operacionais, volatilidade de preços e risco de renda — que tornam a decisão manual sujeita a simplificações e a resultados subótimos. Este trabalho apresenta o desenvolvimento de um sistema web para auxílio ao planejamento de rotações de culturas, implementando o modelo matemático *Target MOTAD* proposto por Salassi, Deliberto e Guidry (2013), que combina uma formulação de fluxo em rede do tipo problema de transporte com o controle paramétrico de risco de renda. O sistema foi calibrado com dez safras históricas (2015/16 a 2024/25) do estado de Goiás, considerando soja, milho e algodão, a partir de dados de produtividade, preço e custo de produção da CONAB e de fatores agronômicos de ajuste de rendimento fundamentados em boletins técnicos da Embrapa. A arquitetura foi estruturada segundo os princípios do *Domain-Driven Design*, com *backend* em C# e .NET 10, *frontend* em Angular 21, banco de dados MariaDB e resolução do modelo de programação linear delegada ao *solver* GLOP do Google OR-Tools. Como resultado, o sistema gera de forma interativa a fronteira eficiente risco-retorno, permitindo ao produtor comparar planos de rotação Pareto-ótimos e selecionar aquele que melhor equilibra renda esperada e tolerância ao risco, sem exigir conhecimento técnico em otimização.

Palavras-chave: Rotação de culturas; programação linear; Target MOTAD; fronteira eficiente; otimização; sistema web.

ABSTRACT

SOUSA, Willian. A web system for crop rotation planning support. Junho, 2026. 40 f. Trabalho de Conclusão de Curso – Bacharel em Ciência da Computação, Instituto Federal Goiano - Campus Rio Verde. Rio Verde, GO, Junho, 2026.

Crop rotation is one of the most relevant agronomic practices for the sustainability of production systems, yet its efficient planning involves multiple interdependent variables — agronomic effects between preceding and succeeding crops, operational constraints, price volatility and income risk — which make manual decision-making prone to simplifications and suboptimal outcomes. This work presents the development of a web system to support crop rotation planning, implementing the *Target MOTAD* mathematical model proposed by Salassi, Deliberto and Guidry (2013), which couples a transportation-type network flow formulation with parametric income-risk control. The system was calibrated with ten historical crop seasons (2015/16 to 2024/25) from the state of Goiás, considering soybean, maize and cotton, based on yield, price and production-cost data from CONAB and on agronomic yield-adjustment factors grounded in Embrapa technical bulletins. The architecture follows Domain-Driven Design principles, with a C#/.NET 10 backend, an Angular 21 frontend, a MariaDB database, and the linear programming model solved by Google OR-Tools' GLOP solver. As a result, the system interactively generates the risk-return efficient frontier, allowing the producer to compare Pareto-optimal rotation plans and select the one that best balances expected income and risk tolerance, without requiring technical knowledge of optimization.

Keywords: crop rotation; linear programming; Target MOTAD; efficient frontier; optimization; web system.

LISTA DE FIGURAS

Figura 1 – Grafo simples.	5
Figura 2 – Grafo direcionado.	5
Figura 3 – Problema simples de rotação de culturas especificado como um modelo de fluxo em rede.	11
Figura 4 – Modelo de fluxo em rede de rotação de culturas modificado.	12
Figura 5 – Tela inicial — resumo de fazendas, culturas e execuções recentes.	22
Figura 6 – Assistente de otimização — Etapa 1.	23
Figura 7 – Assistente de otimização — Etapa 2.	23
Figura 8 – Assistente de otimização — Etapa 3.	24
Figura 9 – Assistente de otimização — Etapa 4.	24
Figura 10 – Resultados — Gráfico de dispersão da fronteira eficiente.	25
Figura 11 – Resultados — Fronteira eficiente e Culturas em rotação.	25

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
GLOP	Google Linear Optimization Package
JSON	JavaScript Object Notation
JWT	JSON Web Token
MySQL	My Structured Query Language
REST	Representational State Transfer
SPD	Sistema de Plantio Direto

LISTA DE ALGORITMOS

Algoritmo 1 – SolveForRisk — Target MOTAD	29
---	----

SUMÁRIO

1 – INTRODUÇÃO	1
2 – REVISÃO DE LITERATURA	2
2.1 Rotação de culturas	2
2.2 Teoria dos grafos	4
2.3 Programação Linear	6
2.4 Problema de transporte	7
2.5 Modelagem de rotação de culturas como problema de fluxo em rede	7
3 – MATERIAIS E MÉTODOS	13
3.1 Linguagens de programação e plataformas de desenvolvimento	13
3.2 Banco de dados e persistência	13
3.3 Biblioteca de otimização: Google OR-Tools	14
3.4 Controle de risco	14
3.5 Metodologia de desenvolvimento	16
4 – DESENVOLVIMENTO	18
4.1 Pesquisa e preparação dos dados	18
4.2 Modelagem do domínio	19
4.3 Implementação das camadas de aplicação e infraestrutura	20
4.4 Interface de programação de aplicação (API REST)	21
4.5 Interface web	22
4.6 Cálculo dos retornos líquidos por par de rotação	24
4.7 Construção e resolução do modelo de programação linear	26
4.8 Integração entre as camadas e fluxo de execução	27
4.9 Algoritmo de Otimização	28
4.10 Código fonte	28
5 – RESULTADOS	30
5.1 Configuração do cenário de teste	30
5.2 Visão geral da interface de resultados	30
5.3 Execução 1 — sem restrição de monocultura	31
5.4 Execução 2 — com restrição de sem monocultura	33
5.5 Execução 3 — dados históricos revisados	34
5.6 Análise comparativa das execuções	35
6 – CONCLUSÃO	37
6.1 Trabalhos futuros	37
Referências	38

1 INTRODUÇÃO

A agricultura brasileira ocupa posição de destaque no cenário global, sendo o Brasil um dos maiores produtores e exportadores de *commodities* agrícolas do mundo. O estado de Goiás, inserido no bioma Cerrado, figura entre os principais polos produtores nacionais, com expressiva participação na produção de soja, milho, algodão e cana-de-açúcar (SILVA et al., 2022). Nesse contexto, a eficiência no uso do solo e a sustentabilidade dos sistemas produtivos são determinantes para a competitividade da agropecuária regional a longo prazo.

A rotação de culturas é uma das práticas agronômicas mais importantes para a manutenção da saúde do solo e da produtividade agrícola sustentável. Ao diversificar as espécies cultivadas em uma mesma área ao longo dos ciclos agrícolas, o produtor interrompe ciclos de pragas e doenças, melhora os atributos físicos, químicos e biológicos do solo e reduz a dependência de insumos agronômicos (FRANCHINI et al., 2011). No Brasil, a rotação de culturas é premissa fundamental do Sistema Plantio Direto (SPD), adotado em mais de 33 milhões de hectares no país (SILVA et al., 2022).

Apesar de seus reconhecidos benefícios, o planejamento eficiente de rotações de culturas é uma tarefa complexa. O produtor precisa considerar simultaneamente a diversidade botânica entre as espécies, os efeitos agronômicos de cada sequência antecessora-sucessora, as restrições operacionais da propriedade, a volatilidade dos preços de mercado e o risco de renda associado a diferentes cenários de produção. Esse conjunto de variáveis interdependentes torna o planejamento manual sujeito a simplificações que podem resultar em sequências subótimas do ponto de vista econômico.

A modelagem do problema de planejamento de rotação como um problema de otimização matemática tem sido explorada na literatura desde os trabalhos de Detlefsen e Jensen (2007), que propuseram a formulação como um modelo de fluxo em rede. Salassi, Deliberto e Guidry (2013) estenderam essa abordagem incorporando restrições laterais e controle de risco de renda pelo modelo *Target MOTAD (Minimization Of Total Absolute Deviation)*, permitindo gerar a fronteira eficiente risco-retorno para o produtor.

Este trabalho apresenta o desenvolvimento de um sistema web para auxílio ao planejamento de rotações de culturas, implementando o modelo matemático de Salassi, Deliberto e Guidry (2013) adaptado para o contexto agrícola do estado de Goiás. O sistema permite que o produtor cadastre suas fazendas e culturas, carregue séries históricas de produtividade, preço e custo de produção, e obtenha, de forma interativa, o conjunto de planos de rotação Pareto-ótimos que maximiza a renda esperada para cada nível de tolerância ao risco.

Do ponto de vista técnico, o sistema foi estruturado segundo os princípios do *Domain-Driven Design (DDD)*, utilizando C# e .NET 10 no *backend*, Angular 21 no *frontend*, MariaDB como banco de dados e o solver GLOP do Google OR-Tools para a resolução do modelo de programação linear.

O presente trabalho apresenta a construção de um sistema de auxílio, para que o produtor possa analisar diversos cenários de rotações de culturas, sem a necessidade de utilizar ferramentas de otimização, em uma interface gráfica amigável e intuitiva que apresenta os resultados e o retorno econômico esperado de cada solução da fronteira eficiente.

2 REVISÃO DE LITERATURA

Para que seja possível construir o sistema de auxílio, é necessário definir o escopo de estudo, desta forma, contextualizando o que será abordado e o que será desenvolvido. Este capítulo tem como objetivo apresentar conceitos e definições existentes na literatura.

2.1 Rotação de culturas

A rotação de culturas consiste na alternância sistemática de diferentes espécies vegetais em uma mesma área agrícola ao longo do tempo. Esta prática é recomendada com o objetivo de evitar a sucessão contínua da mesma cultura no mesmo talhão, o que ameaça a sustentabilidade do sistema produtivo pela exaustão dos recursos do solo e pelo acúmulo de patógenos, pragas e plantas daninhas adaptados a uma única espécie (FRANCHINI et al., 2011).

É importante distinguir a rotação de culturas do sistema de sucessão de culturas. Na sucessão, duas espécies se alternam de forma fixa e repetida a cada ano — como ocorre no sistema soja/milho safrinha — sem variação na sequência ao longo dos ciclos. Na rotação, há um planejamento com maior diversidade botânica ao longo de ciclos plurianuais, incorporando espécies de diferentes famílias, com distintos sistemas radiculares, exigências nutricionais e hábitos de crescimento (Embrapa, 2021; CASTELLAZZI et al., 2008).

De acordo com Castellazzi et al. (2008), a representação sistemática de uma rotação de culturas deve considerar quatro relações fundamentais: o período mínimo para que uma mesma cultura retorne ao mesmo talhão; os efeitos — benéficos ou prejudiciais — que uma cultura exerce sobre a seguinte; os ciclos de plantio e colheita dentro de um mesmo ano; e as proporções de área destinadas a cada cultura na propriedade.

A rotação de culturas é amplamente reconhecida como uma das práticas mais relevantes para a sustentabilidade dos sistemas agrícolas intensivos. No Brasil, ela constitui premissa fundamental do Sistema Plantio Direto (SPD), que alcança mais de 33 milhões de hectares cultivados no país (SILVA et al., 2022). Do ponto de vista agrônomo, o cultivo contínuo de uma mesma espécie favorece a proliferação de patógenos, insetos-praga e plantas daninhas adaptados àquela cultura, elevando os custos de produção com defensivos e reduzindo a produtividade ao longo dos anos (FRANCHINI et al., 2011).

Do ponto de vista ambiental, a rotação contribui para a conservação do solo ao promover o aporte de resíduos vegetais de diferentes composições químicas, favorecendo a atividade microbiana, o sequestro de carbono e a ciclagem eficiente de nutrientes. Plantas com arquiteturas radiculares distintas atuam em diferentes profundidades do perfil do solo, reduzindo a compactação, ampliando a macroporosidade e melhorando a disponibilidade de água e nutrientes para as culturas subsequentes (SILVA et al., 2022).

Os benefícios da rotação de culturas manifestam-se nos atributos físicos, químicos e biológicos do solo, bem como na produtividade e na rentabilidade da propriedade.

No que diz respeito à fertilidade do solo, a inclusão de leguminosas — como soja, crotalária ou guandu — na sequência de rotação proporciona a fixação biológica de nitrogênio (FBN). Essas espécies se associam a bactérias do gênero *Rhizobium*, capazes de captar o nitrogênio da atmosfera e disponibilizá-lo no solo, reduzindo a necessidade de adubação nitrogenada para a cultura sucessora. Estudos demonstraram incremento de até 539 kg/ha na produtividade do milho para cada ano de soja cultivada como antecessora (Embrapa, 2021).

A rotação também promove melhoria das propriedades físicas do solo. Raízes de diferentes arquiteturas — pivotantes, fasciculadas, profundas ou superficiais — atuam em distintas camadas, reduzindo a compactação e aumentando a macroporosidade, o que favorece a infiltração de água, a aeração e o desenvolvimento do sistema radicular das culturas seguintes (SILVA et al., 2022).

Quanto ao controle fitossanitário, espécies de famílias botânicas distintas não hospedam os mesmos patógenos, interrompendo os ciclos de doenças e reduzindo a pressão de pragas e plantas daninhas. Leandro e Asmus (2015) demonstraram que a rotação de milho, crotalária ou soja com braquiária como cobertura reduziu significativamente a densidade populacional do nematoide *Rotylenchulus reniformis*, com reflexos positivos sobre a produtividade da soja na safra seguinte.

Do ponto de vista econômico, a rotação de culturas pode aumentar a produção de alimentos em até 65% em comparação ao monocultivo contínuo, além de reduzir os custos unitários de produção pela menor dependência de insumos agrícolas. A longo prazo, a diversificação também confere maior estabilidade ao sistema produtivo frente às oscilações de mercado (FRANCHINI et al., 2011).

No Brasil, a rotação de culturas ganhou relevância crescente a partir da disseminação do SPD, que se iniciou na Região Sul nos anos 1970 e chegou ao Cerrado na década de 1990 com os fluxos migratórios de agricultores sulistas experientes nessa prática (SILVA et al., 2022). A Empresa Brasileira de Pesquisa Agropecuária (Embrapa) desempenhou papel central na adaptação e na recomendação de sistemas de rotação para as distintas condições edafoclimáticas do país, desenvolvendo cultivares e protocolos técnicos para cada região.

Apesar dos avanços, o Cerrado ainda é caracterizado predominantemente pelo monocultivo de soja ou pelo sistema contínuo de sucessão soja/milho safrinha. Esse cenário tende a provocar degradação física, química e biológica do solo, além de favorecer condições mais propícias ao desenvolvimento de doenças, pragas e plantas daninhas, comprometendo a produtividade e a sustentabilidade do sistema a longo prazo (SILVA et al., 2022). Estudos realizados pela Embrapa demonstraram que diversas rotações de culturas testadas no Cerrado proporcionam aumento do rendimento econômico, com destaque para as rotações entre espécies da família Poaceae (gramíneas) e Fabaceae (leguminosas) (SILVA et al., 2022).

O estado de Goiás está inserido no bioma Cerrado e figura entre os principais polos de produção agropecuária do Brasil. A soja (*Glycine max* L.) é a cultura de verão predominante, ocupando vastas áreas do estado. A seleção de espécies para compor um sistema de rotação nessa região deve basear-se na diversidade botânica, priorizando plantas com diferentes sistemas radiculares, hábitos de crescimento e exigências nutricionais, de modo a interromper ciclos de pragas e doenças e a reduzir custos de produção (Embrapa, 2021).

As principais culturas utilizadas em rotação no estado de Goiás são:

- **Milho** (*Zea mays* L.): principal cultura de rotação e sucessão com a soja, amplamente cultivado na modalidade de segunda safra (safrinha), com elevado potencial de produção de palhada e interrupção dos ciclos de doenças da soja;
- **Sorgo granífero** (*Sorghum bicolor* L.): Goiás é o maior produtor nacional desta cultura, que se destaca pela tolerância a déficits hídricos e pela maior amplitude de épocas de plantio sem perdas severas de produtividade, tornando-a especialmente atrativa como alternativa ao milho safrinha em anos com menor disponibilidade hídrica;
- **Milheto** (*Pennisetum glaucum* L.): amplamente utilizado como cultura de cobertura na entressafra, com alta produção de biomassa, supressão de plantas daninhas e

proteção do solo contra erosão;

- **Braquiária** (*Urochloa* spp.): gramínea forrageira empregada tanto em sistemas de integração lavoura-pecuária quanto como planta de cobertura em consórcio com milho safrinha, contribuindo para a melhoria dos atributos físicos do solo;
- **Leguminosas de cobertura** — como crotalária (*Crotalaria* spp.) e guandu (*Cajanus cajan* L.) —: utilizadas como adubos verdes, com contribuição relevante para a fixação biológica de nitrogênio e a supressão de nematoides;
- **Girassol** (*Helianthus annuus* L.): utilizado em menor escala, recomendado como opção de rotação pela profundidade de seu sistema radicular e resistência moderada à seca.

A Embrapa recomenda que, em cada talhão da região central do Brasil, a soja seja cultivada por no máximo dois anos consecutivos, seguida por pelo menos dois anos com outras espécies — como milho, arroz, algodão ou sorgo —, em um ciclo plurianual que preserve a saúde do solo e a sustentabilidade do sistema produtivo (Embrapa, 2021).

2.2 Teoria dos grafos

A teoria dos grafos é um ramo da matemática que estuda as propriedades de grafos, estruturas abstratas utilizadas para modelar relações entre objetos (NETTO, 2003). Originada no século XVIII a partir do trabalho de Leonhard Euler sobre o problema das Sete Pontes de Königsberg, a teoria dos grafos evoluiu para uma das ferramentas matemáticas mais versáteis da ciência moderna, encontrando aplicações em ciência da computação, engenharia, biologia, economia, logística e muitas outras áreas. Em particular, o problema de planejamento de rotação de culturas pode ser naturalmente formulado como um problema de fluxo em rede, tornando os conceitos desta teoria fundamentais para o desenvolvimento deste trabalho.

Formalmente, um grafo $G = (V, E)$ é composto por um conjunto finito de vértices (também denominados nós) V e um conjunto de arestas (também chamadas de arcos ou linhas) $E \subseteq V \times V$, que conectam pares de vértices (SZWARCFITER, 2018). As arestas representam as relações ou conexões entre os objetos modelados pelos vértices, podendo ser ponderadas — quando possuem um valor numérico (peso) associado — ou não ponderadas. A Figura 1 exemplifica um grafo simples.

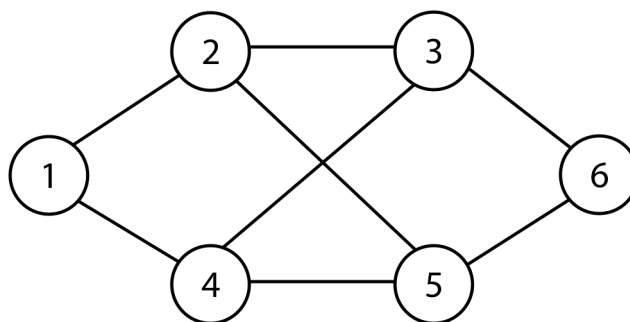
Dado um grafo G , define-se a conectividade como a existência de um caminho que conecta um vértice a outro. Um grafo é dito **conexo** quando existe pelo menos um caminho entre quaisquer dois vértices do conjunto V . Um grafo é dito **fortemente conexo** quando, para todo par de vértices $u, v \in V$, existe um caminho de u até v e outro de v até u (GOLDBARG; GOLDBARG, 2012).

Grafos direcionados

Um **grafo direcionado** (ou dígrafo) é um tipo de grafo em que cada aresta possui uma direção definida, sendo orientada de um vértice de origem para um vértice de destino. Nesse caso, as arestas são denominadas **arcos**, representando relações unidirecionais entre os vértices (GOLDBARG; GOLDBARG, 2012). Formalmente, um dígrafo $G = (V, A)$ é composto por um conjunto de vértices V e um conjunto de arcos $A \subseteq V \times V$, onde cada arco $(u, v) \in A$ indica que existe uma conexão orientada do vértice u para o vértice v , mas não necessariamente de v para u .

Em um dígrafo, define-se o **grau de entrada** (*in-degree*) de um vértice v como o número de arcos que chegam a v , e o **grau de saída** (*out-degree*) como o número de arcos

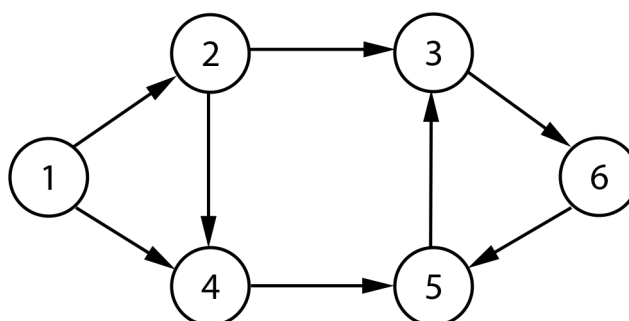
Figura 1 – Grafo simples.



Fonte: próprio autor (2026).

que partem de v . Essa distinção é essencial em aplicações onde a direção do fluxo possui significado prático, como em redes de transporte, cadeias de suprimentos e, conforme será abordado nas seções seguintes, no planejamento de sequências de culturas agrícolas (NETTO, 2003). A Figura 2 a seguir representa um grafo direcionado.

Figura 2 – Grafo direcionado.



Fonte: próprio autor (2026).

O **fluxo em redes** é um conceito fundamental da teoria dos grafos que modela a transferência de recursos — como mercadorias, informações, energia ou capital — através de uma rede de caminhos (GOLDBARG; GOLDBARG, 2012). Uma rede de fluxo é representada por um dígrafo ponderado $G = (V, A)$, onde cada arco $(u, v) \in A$ possui uma **capacidade** $c(u, v) \geq 0$ que representa a quantidade máxima de recurso que pode ser transportada por aquele arco.

A rede inclui dois nós especiais: o **nó de origem** s (*source*), de onde parte o fluxo, e o **nó de destino** t (*sink*), onde o fluxo é consumido. Os demais nós são denominados nós de **transbordo**, nos quais o fluxo que entra deve ser igual ao fluxo que sai — propriedade conhecida como **lei de conservação do fluxo** (NETTO, 2003). Formalmente, para todo nó intermediário $v \in V \setminus \{s, t\}$, deve-se satisfazer:

$$\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w) \quad (1)$$

onde $f(u, v)$ denota o fluxo pelo arco (u, v) , sujeito à restrição de capacidade $0 \leq f(u, v) \leq c(u, v)$.

O problema clássico de fluxo em redes consiste em determinar a quantidade máxima de recurso que pode ser transferida da origem s ao destino t , respeitando as capacidades dos arcos. Este problema, conhecido como **problema do fluxo máximo**, possui algoritmos eficientes de resolução, como o algoritmo de Ford-Fulkerson e suas variantes (GOLDBARG; GOLDBARG, 2012).

2.3 Programação Linear

A programação linear é um ramo da otimização matemática que trata da maximização ou minimização de uma função objetivo linear sujeita a um conjunto finito de restrições igualmente lineares (DANTZIG, 2002; BAZARAA; JARVIS; SHERALI, 2011). É uma das ferramentas mais difundidas da pesquisa operacional, com aplicações em finanças, logística, engenharia, economia e planejamento agrícola (HILLIER, 2005).

Um problema de programação linear (PPL) pode ser escrito na forma padrão:

$$\text{Maximizar (ou Minimizar)} \quad z = c^T x \quad (1)$$

$$\text{sujeito a} \quad Ax \leq b \quad (2)$$

$$x \geq 0, \quad (3)$$

em que $x \in \mathbb{R}^n$ é o vetor de *variáveis de decisão*; $c \in \mathbb{R}^n$ é o vetor de *coeficientes de custo/lucro* da função objetivo; $A \in \mathbb{R}^{m \times n}$ é a *matriz de restrições*; e $b \in \mathbb{R}^m$ é o vetor de *recursos disponíveis* (os termos independentes das restrições).

O conjunto de pontos que satisfazem todas as restrições é denominado *região factível*. Quando não vazia e limitada, essa região forma um *poliedro convexo*, e um resultado fundamental da teoria garante que, se existe solução ótima, então ao menos uma delas ocorre em um *vértice* (solução básica factível) desse poliedro (BAZARAA; JARVIS; SHERALI, 2011). Essa propriedade é o que torna o problema computacionalmente tratável: em vez de examinar infinitos pontos, basta percorrer um número finito de vértices.

O **Método Simplex**, proposto por Dantzig (DANTZIG, 2002), é o algoritmo mais amplamente utilizado para resolver PPLs. Ele opera sobre as soluções básicas factíveis (os vértices do poliedro), partindo de uma solução inicial e migrando iterativamente para vértices adjacentes que melhorem o valor da função objetivo, até que nenhum vizinho ofereça ganho — condição que caracteriza a otimalidade (DANTZIG, 2002; BAZARAA; JARVIS; SHERALI, 2011). O *solver* GLOP, do Google OR-Tools, empregado neste trabalho, é uma implementação moderna do Método Simplex para problemas de programação linear contínua.

A importância da programação linear para este trabalho é dupla. Primeiro, o *problema de transporte* (Seção 2.4) e a *modelagem de rotação de culturas* (Seção 2.5) são casos particulares de PPL. Segundo, o modelo *Target MOTAD* adotado, mesmo incorporando o controle de risco, permanece **integralmente linear**, o que permite resolvê-lo de forma exata e eficiente pelo Método Simplex, sem recorrer a técnicas de programação quadrática.

2.4 Problema de transporte

O **problema do transporte** é uma extensão clássica do fluxo em redes, voltada para a otimização da distribuição de recursos de um conjunto de origens (fontes) para um conjunto de destinos (demandas), com o objetivo de minimizar o custo total de transporte (BAZARAA; JARVIS; SHERALI, 2011). Em termos de grafos, o problema é representado por um grafo bipartido, onde os vértices são divididos em dois conjuntos disjuntos: o conjunto das fontes S e o conjunto dos destinos D . Cada fonte $i \in S$ possui uma oferta s_i de recursos disponíveis, e cada destino $j \in D$ possui uma demanda d_j a ser atendida. O custo de transportar uma unidade de recurso da fonte i ao destino j é dado por c_{ij} (PRADO, 2016).

O modelo matemático do problema do transporte é formulado como:

$$\text{Minimizar } Z = \sum_{i \in S} \sum_{j \in D} c_{ij} \cdot x_{ij} \quad (1)$$

sujeito a:

$$\sum_{j \in D} x_{ij} \leq s_i, \quad \forall i \in S \quad (\text{restrição de oferta}) \quad (2)$$

$$\sum_{i \in S} x_{ij} \geq d_j, \quad \forall j \in D \quad (\text{restrição de demanda}) \quad (3)$$

$$x_{ij} \geq 0, \quad \forall i \in S, j \in D \quad (4)$$

onde x_{ij} representa a quantidade de recurso transportada da fonte i ao destino j .

O problema do transporte é relevante em diversas áreas, como logística, distribuição de produtos, planejamento de rotas e design de redes de transporte (BAZARAA; JARVIS; SHERALI, 2011). Sua estrutura também serve de base para a modelagem do problema de rotação de culturas, como será detalhado nas seções seguintes.

2.5 Modelagem de rotação de culturas como problema de fluxo em rede

A representação do problema de rotação de culturas como um modelo de fluxo em rede foi proposta por Detlefsen e Jensen (2007), que desenvolveram um modelo plurianual de otimização de fluxo máximo capaz de incorporar anos de safra anteriores e anos de produção futuros. Nesta formulação, as culturas disponíveis para plantio em cada ano são representadas como nós intermediários da rede, enquanto os arcos conectam culturas que podem se suceder, com capacidades correspondentes às áreas de plantio viáveis. A Figura 3 ilustra esta estrutura.

Embora eficiente, o modelo de Detlefsen e Jensen (2007) apresenta limitações: a formulação simples de rede não incorpora restrições laterais sobre áreas de cultivo específicas e não considera o impacto do preço de mercado nem o risco de rendimento na escolha das sequências ideais.

Salassi, Deliberto e Guidry (2013) estenderam este modelo para suportar o planejamento de dois anos de rotação com n culturas, adicionando restrições laterais que aproximam a formulação da realidade do planejamento agrícola. O modelo modificado, ilustrado na Figura 4, faz uso da estrutura do problema do transporte, representando as relações de sucessão entre culturas por meio de um grafo bipartido ponderado.

O modelo matemático adotado neste trabalho é baseado na formulação de Salassi, Deliberto e Guidry (2013), que define como função objetivo a maximização do retorno líquido de um conjunto de n culturas em rotação ao longo de dois anos de produção. O modelo considera dados históricos de mercado e um índice de risco de produção associado a cada sequência de culturas.

A lista atual mistura, sob o rótulo único de “variáveis de decisão”, elementos de naturezas diferentes: parâmetros (dados de entrada, como RL , TFA e p_k) aparecem ao lado das verdadeiras variáveis de decisão (x e Y_k). A separação correta é a seguinte.

Conjuntos.

$$\mathcal{C} = \{1, \dots, n\}$$

conjunto das n culturas disponíveis para o planejamento (ex.: soja, milho, algodão).

$$\mathcal{A} = \{1, 2\}$$

conjunto dos dois anos de produção que compõem o horizonte da rotação.

$$\mathcal{K} = \{1, \dots, m\}$$

conjunto dos m cenários históricos (cada cenário é um ano da série histórica de preços e produtividades).

Índices.

$i \in \mathcal{C}$ cultura *antecessora* (plantada no ano 1).

$j \in \mathcal{C}$ cultura *sucessora* (plantada no ano 2).

$k \in \mathcal{K}$ cenário histórico (ano da série histórica).

Parâmetros (dados de entrada, conhecidos antes da otimização).

$RL_{i,1,j}$ retorno líquido esperado (R\$/ha) da cultura i no ano 1 quando sucedida por j .

$RL_{j,2,i}$ retorno líquido esperado (R\$/ha) da cultura j no ano 2 quando sucede i .

$RL_{i,1,j,k}$ retorno líquido *histórico* (R\$/ha) da sequência $i \rightarrow j$ no cenário k (análogo para $RL_{j,2,i,k}$).

TFA área total de plantio da propriedade, em hectares..

T renda-alvo da propriedade (R\$) — nível de renda de referência abaixo do qual se mede o risco.

p_k probabilidade de ocorrência do cenário k ; adota-se $p_k = 1/m$ (cenários equiprováveis).

L, K_i limites das restrições laterais.

Variáveis de decisão (determinadas pelo *solver*).

$x_{i,1,j} \geq 0$ área (ha) alocada à sequência $i \rightarrow j$ no ano 1.

$x_{j,2,i} \geq 0$ área (ha) alocada à mesma sequência no ano 2.

$Y_k \geq 0$ variável auxiliar de *desvio*: mede o quanto a renda do plano, avaliada no cenário histórico k , fica *abaixo* da renda-alvo T (déficit, em R\$).

Modelo matemático

A função objetivo que maximiza o retorno líquido total é dada por:

$$\text{Maximizar } P = \sum_{i=1}^n \sum_{j=1}^n RL_{i,1,j} x_{i,1,j} + \sum_{i=1}^n \sum_{j=1}^n RL_{j,2,i} x_{j,2,i} \quad (1)$$

Sujeito às seguintes restrições:

$$\sum_{i=1}^n \sum_{j=1}^n x_{i,1,j} = TFA \quad (2)$$

$$x_{i,1,j} - x_{j,2,i} = 0, \quad \forall i = 1, \dots, n \text{ e } j = 1, \dots, n \quad (3)$$

$$\sum_{j=1}^n \sum_{i=1}^n x_{j,2,i} = TFA \quad (4)$$

$$x_{i,k,j} - x_{j,k,i} = 0, \quad \forall i \neq j \text{ e para todo } k \quad (5)$$

$$0 \leq x_{ijk} \leq L, \quad \forall i \neq j \text{ e para todo } k \quad (6)$$

$$0 \leq \sum_{j=1}^n x_{i,1,j} \leq K_i, \quad \forall i = 1, \dots, n \quad (7)$$

$$\sum_{i=1}^n \sum_{j=1}^n RL_{i,1,j,k} x_{i,1,j} + \sum_{i=1}^n \sum_{j=1}^n RL_{j,2,i,k} x_{j,2,i} + Y_k \geq T, \quad \forall k = 1, \dots, m \quad (8)$$

$$\sum_{k=1}^m p_k Y_k = \theta, \quad \text{onde } \theta \text{ varia do seu valor máximo até zero} \quad (9)$$

$$\theta = D = M/2 \quad (10)$$

Restrições laterais: equações (6) e (7)

As equações (7) e (8) são *restrições laterais* opcionais, que permitem ao produtor impor condições de mercado ou operacionais ao plano. Elas se distinguem pelo *escopo*:

- **Equação (6) — limite por sequência de rotação:** Limita a área de *uma sequência específica* $i \rightarrow j$. Por exemplo, “a área da rotação soja \rightarrow algodão não pode exceder $L = 50$ ha”. O operador \leq define um *teto* (limite superior) e o operador \geq , um *piso* (limite inferior).
- **Equação (7) — limite por cultura:** Limita a área *total* dedicada a uma cultura i no ano 1, *somada sobre todas as sucessoras* j . Por exemplo, “a área total de soja deve ser de *no mínimo* $K_i = 200$ ha, independentemente de qual cultura a sucede”. Enquanto a equação (6) atua sobre um único par de rotação, a equação (7) atua sobre o conjunto de pares que partem de uma mesma cultura.

Controle de risco: equações (8), (9) e (10)

O bloco de equações (8)–(10) implementa o *Target MOTAD* (*Target Minimization Of Total Absolute Deviation*), formulação de programação linear para risco proposta por Hazell (1971) e por Tauer (1983), e adotada por Salassi, Deliberto e Guidry (2013). A ideia central é medir o risco de um plano pelos *déficits de renda* que ele teria sofrido se cada ano histórico se repetisse, e então *limitar* esses déficits.

Equação (8) — mensuração do déficit em cada cenário.

Para *cada* cenário histórico k , a expressão calcula a renda que o *mesmo* plano de áreas x teria gerado se os preços e produtividades fossem os efetivamente observados naquele ano. A variável Y_k “completa” a diferença até o alvo T : se a renda do cenário k atinge ou supera T , então $Y_k = 0$; se fica abaixo, $Y_k = T - (\text{renda no cenário } k)$ representa exatamente o tamanho do *déficit*. Há uma restrição dessas para cada um dos m anos históricos.

Equação (9) — medida agregada de risco.

isto é, o risco do plano é o *déficit médio esperado* sobre todos os cenários. Com cenários equiprováveis ($p_k = 1/m$), θ é simplesmente a média aritmética dos déficits Y_k . O parâmetro θ é *paramétrico*: ele é variado de um valor máximo até zero para gerar diferentes planos. Na implementação, essa equação aparece como a desigualdade $\frac{1}{m} \sum_k Y_k \leq h$, que limita o risco do plano ao teto h .

Equação (10) — por que apenas os déficits.

Esta relação justifica medir o risco usando *somente* os déficits Y_k (desvios *abaixo* do alvo). Para uma série de retornos em torno de uma média, a soma dos desvios negativos é igual à soma dos desvios positivos; logo, o desvio absoluto médio total M reparte-se em duas metades iguais, e a parcela negativa (o déficit) corresponde a $D = M/2$. Em termos práticos: como o lado “ruim” (renda abaixo do alvo) e o lado “bom” (renda acima) são simétricos, basta controlar um deles. O modelo controla o lado ruim, que é o que de fato interessa ao produtor avesso ao risco.

Síntese do mecanismo.

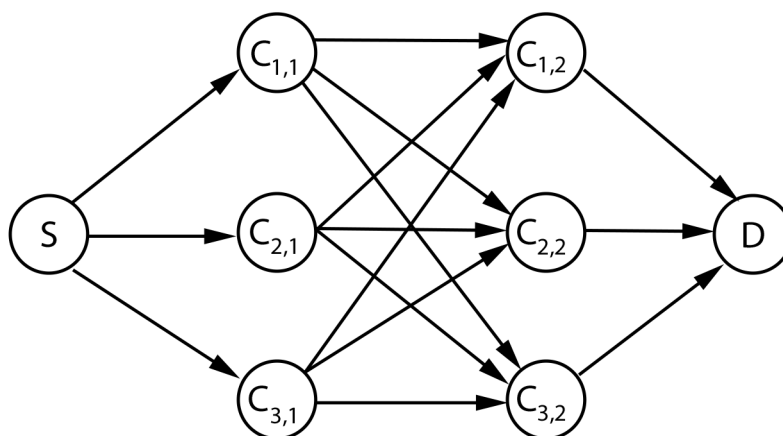
Em conjunto, (8)–(10) funcionam assim: a equação (8) *mede* quanto cada ano histórico ficaria abaixo do alvo; a equação (9) *agrega* esses déficits em um único número θ (ou h); e variar esse número do máximo até zero produz uma família de planos que vai do mais arrojado (tolera maiores déficits em troca de maior renda esperada) ao mais conservador (não admite ficar abaixo do alvo em nenhum cenário). Cada plano é um ponto da *fronteira eficiente risco-retorno*.

Limitantes inferiores e superiores do modelo

O modelo combina três tipos de limitantes:

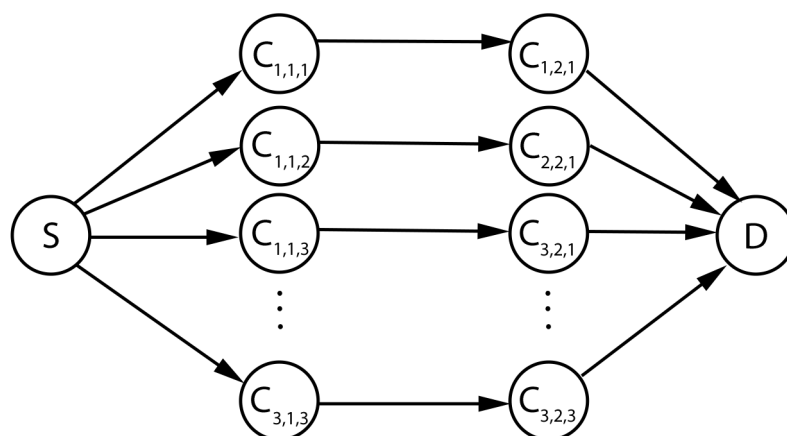
1. **Limitantes de não-negatividade (piso natural).** Todas as variáveis são não-negativas: $x_{i,1,j} \geq 0$, $x_{j,2,i} \geq 0$ e $Y_k \geq 0$. Áreas e déficits não podem ser negativos — não existe “área negativa” nem “déficit negativo” (um superávit não reduz o risco no modelo).
2. **Limitantes laterais (definidos pelo usuário).** As equações (6) e (7) introduzem limites *inferiores* ($\geq L$, $\geq K_i$) ou *superiores* ($\leq L$, $\leq K_i$) sobre áreas de sequências ou de culturas. São opcionais e refletem políticas de mercado ou restrições operacionais (ex.: contrato mínimo de soja; área máxima de algodão por capacidade de colheita). Observe-se que a área *total* não é um intervalo, e sim uma *igualdade*: as equações (2) e (4) fixam a soma das áreas em exatamente TFA .
3. **Limitante do risco (gera a fronteira).** O nível de risco admissível é limitado superiormente por h (equação 9). Esse h varia no intervalo $[0, h_{\max}]$, em que h_{\max} é o *limite superior* natural do risco (o risco do plano de máximo retorno, sem restrição) e 0 é o *limite inferior* (nenhum déficit tolerado). Valores de h abaixo de um certo ponto tornam o problema *infectível* — é impossível atingir o alvo T em todos os cenários simultaneamente — e esses pontos são descartados da fronteira. Este modelo constitui a base matemática do sistema web desenvolvido neste trabalho, cuja implementação é detalhada no Capítulo 4.

Figura 3 – Problema simples de rotação de culturas especificado como um modelo de fluxo em rede.



Fonte: Salassi, Deliberto e Guidry (2013).

Figura 4 – Modelo de fluxo em rede de rotação de culturas modificado.



Fonte: Salassi, Deliberto e Guidry (2013).

3 MATERIAIS E MÉTODOS

O sistema é separado em duas partes, sendo elas o *backend* responsável pela modelagem dos dados, pela lógica de negócio e pela execução do modelo de otimização, e por um *frontend* interativo que expõe as funcionalidades ao usuário final por meio de uma interface gráfica acessível via navegador.

3.1 Linguagens de programação e plataformas de desenvolvimento

O *backend* do sistema foi desenvolvido em C# sobre a plataforma *open source* .NET 10, disponibilizada pela Microsoft. O .NET 10 constitui a versão de suporte de longo prazo (LTS) da plataforma, mantida com atualizações de segurança e correções por três anos a partir de seu lançamento (Microsoft, 2024b). Sua escolha foi motivada pela maturidade do ecossistema, pelo suporte nativo a contextos de alta concorrência via *async/await*, pela tipagem estática que favorece a detecção precoce de erros e pelo robusto sistema de injeção de dependências embutido no *framework* ASP.NET Core.

O componente web do *backend* utiliza o *framework* ASP.NET Core Web API, que provê infraestrutura para a criação de serviços *REST*, incluindo roteamento, negociação de conteúdo, *middleware* e integração com ferramentas de autenticação e documentação. A API expõe seus recursos sob o prefixo de versão `/api/v1/`, adotando o padrão de versionamento de API para garantir compatibilidade com clientes existentes ao longo da evolução do sistema.

O *frontend* foi implementado em TypeScript 5.9 utilizando o *framework* Angular 21 (Google LLC, 2025). O TypeScript é um superconjunto tipado do JavaScript que adiciona verificação estática de tipos, interfaces e decoradores, o que aumenta a robustez do código e facilita a manutenção em projetos de médio e grande porte.

O Angular é um *framework* de código aberto mantido pela Google para o desenvolvimento de aplicações *Single-Page Application* (SPA). Na versão 21 foram adotados os *standalone components*, modelo que elimina a necessidade de NgModules e permite declarar o escopo de cada componente de forma independente. O carregamento preguiçoso (*lazy loading*) de rotas foi empregado para reduzir o tamanho do pacote inicial transmitido ao navegador, melhorando o tempo de carregamento da aplicação.

A interface gráfica adota a especificação Material Design por meio da biblioteca Angular Material, que fornece um conjunto coeso de componentes visuais seguindo as diretrizes de acessibilidade e usabilidade propostas pela Google (Google LLC, 2024). A visualização dos resultados da otimização faz uso da biblioteca Chart.js, integrada ao Angular por meio do pacote ng2-charts, para renderizar o gráfico da fronteira eficiente de risco-retorno.

3.2 Banco de dados e persistência

O sistema de gerenciamento de banco de dados relacional adotado foi o MariaDB 11 (MariaDB Foundation, 2023), *fork open source* do MySQL mantido pela MariaDB Foundation. A escolha é justificada pela compatibilidade total com o dialeto SQL do MySQL, pela licença de código aberto, pela disponibilidade de imagem oficial no Docker Hub e pelo suporte estável ao driver de acesso a dados utilizado pelo ORM da aplicação.

O banco armazena todas as entidades do domínio: usuários, propriedades rurais, culturas agrícolas, dados históricos de produção, ajustes de rendimento por rotação, execuções de otimização, soluções e alocações de área. As tabelas foram geradas e versionadas por meio de migrações controladas pelo ORM, garantindo rastreabilidade das alterações no esquema ao longo do ciclo de desenvolvimento.

O mapeamento objeto-relacional é realizado pelo Entity Framework Core (EF Core) 9.0, ORM oficial da plataforma .NET (Microsoft, 2024a). EF Core adota o modelo *code-first*: as tabelas e relacionamentos são definidos como classes C# e o esquema do banco é gerado automaticamente a partir dessas definições. O provedor de banco de dados utilizado é o *Pomelo.EntityFrameworkCore.MySql*, projeto de código aberto que implementa o suporte ao MariaDB/MySQL para o EF Core.

As configurações de mapeamento foram definidas por meio de classes que implementam a interface `IEntityTypeConfiguration<T>`, separando as responsabilidades de configuração do banco de dados das classes de domínio e mantendo o modelo de domínio livre de dependências de infraestrutura. O *framework* executa as migrações pendentes automaticamente na inicialização da aplicação, garantindo que o esquema do banco esteja sempre sincronizado com o modelo de domínio.

3.3 Biblioteca de otimização: Google OR-Tools

A resolução do modelo matemático de otimização é delegada ao Google OR-Tools 9.15 (Google LLC, 2023), biblioteca de código aberto desenvolvida e mantida pela Google para problemas de otimização combinatória e contínua. O OR-Tools disponibiliza diversos *solvers*, dentre os quais o *Google Linear Optimization Package* (GLOP), um solver de programação linear baseado no método Simplex que foi selecionado para este trabalho.

O GLOP é invocado por meio da classe `Solver` da API C# do OR-Tools. Para cada ponto da fronteira eficiente, instancia-se um problema de programação linear com as variáveis de decisão, as variáveis auxiliares de desvio de renda e as restrições do modelo Target MOTAD. Ao término da resolução, o valor da função objetivo e os valores das variáveis de decisão são extraídos e mapeados para as entidades de domínio que representam a solução.

3.4 Controle de risco

Para controlar o risco, é utilizada a variável denominada h_{\max} , ela representa o *maior nível de risco útil* do problema. Ele é obtido resolvendo-se o modelo **sem** a restrição de limite de risco (equação 8): o *solver* encontra então o plano de *máximo retorno esperado* — que é também o de maior risco — e calcula-se o nível de risco desse plano, $h_{\max} = \frac{1}{m} \sum_k Y_k^*$. Acima de h_{\max} a restrição de risco fica inativa e a solução não muda; por isso h_{\max} é o teto a partir do qual a varredura começa.

Para traçar a fronteira eficiente, o intervalo $[0, h_{\max}]$ é dividido em n valores *igualmente espaçados*, com passo

$$\Delta h = \frac{h_{\max}}{n-1}, \quad h_t = h_{\max} \left(1 - \frac{t}{n-1} \right), \quad t = 0, 1, \dots, n-1.$$

Para cada valor h_t , um *problema de programação linear independente* é montado (agora com a restrição $\frac{1}{m} \sum_k Y_k \leq h_t$) e resolvido pelo GLOP, produzindo *um ponto* da fronteira eficiente. Quanto **maior** n , mais densa e suave a fronteira (mais planos para o produtor

comparar); quanto **menor**, mais grosseira. Os pontos em que o LP resulta infactível (risco baixo demais) são descartados, restando apenas as soluções Pareto-ótimas. No sistema, n corresponde ao “número de pontos na fronteira” informado pelo usuário no assistente de criação da otimização.

Para tornar concreto o funcionamento do modelo — em especial o controle de risco —, considere um exemplo reduzido com **duas culturas** (soja S e milho M), área total $TFA = 100$ ha e **três cenários históricos** ($m = 3$, $p_k = 1/3$). Quatro sequências são admissíveis: as monoculturas $S \rightarrow S$ e $M \rightarrow M$ e as rotações $S \rightarrow M$ e $M \rightarrow S$. A Tabela 1 traz o retorno líquido (R\$/ha, somando os dois anos) de cada sequência em cada cenário.

Tabela 1 – Retornos líquidos por hectare (R\$/ha) fictícios.

Sequência	Cen. 1	Cen. 2	Cen. 3	Média NR
$S \rightarrow S$ (monocultura)	5 200	3 000	6 200	4 800
$S \rightarrow M$ (rotação)	4 800	4 200	5 100	4 700
$M \rightarrow S$ (rotação)	4 600	4 000	4 900	4 500
$M \rightarrow M$ (monocultura)	3 800	1 800	4 600	3 400

Fonte: próprio autor (2026).

Adote a renda-alvo $T = 420\,000$ (equivalente a R\$ 4 200/ha sobre os 100 ha). A monocultura de soja tem a *maior renda média* (R\$ 4 800/ha), mas é *volátil*: no cenário 2 rende apenas R\$ 3 000/ha. A rotação $S \rightarrow M$ rende um pouco menos na média (R\$ 4 700/ha), porém é mais *estável* (nunca abaixo de R\$ 4 200/ha). É esse *trade-off* que o controle de risco explora.

Passo 1 — determinar h_{\max} .

Resolvendo o modelo *sem* limite de risco, o *solver* escolhe o plano de maior renda média: 100 ha em $S \rightarrow S$, com renda esperada $P = 480\,000$. Avaliando esse plano nos três cenários, apenas o cenário 2 fica abaixo do alvo: $Y_2 = 420\,000 - 3\,000 \times 100 = 120\,000$ (os demais $Y_k = 0$). Logo

$$h_{\max} = \frac{1}{3}(0 + 120\,000 + 0) = 40\,000.$$

Passo 2 — varrer h de h_{\max} até 0 em $n = 5$ passos.

Para cada h_t resolve-se um LP independente. A Tabela 2 mostra a fronteira eficiente resultante.

Leitura do resultado.

À medida que h *diminui* (menor tolerância ao déficit), o plano migra gradualmente da monocultura de soja — de maior renda esperada (R\$ 480 000) e maior risco — para a rotação $S \rightarrow M$ — de menor renda esperada (R\$ 470 000), porém sem nenhum déficit ($h = 0$): mesmo no pior cenário histórico, a rotação atinge o alvo. A diferença de renda esperada entre os extremos (R\$ 10 000, cerca de 2%) é o “preço” da segurança. Cada linha da Tabela 2 é um ponto Pareto-ótimo da fronteira: nenhum outro plano oferece maior renda para aquele nível de risco. Note ainda que, se a renda-alvo fosse mais exigente, os

Tabela 2 – Fronteira eficiente do exemplo ($n = 5$ passos equiespaçados).

Passo	h (limite)	P (R\$)	Plano ótimo (ha)
1	40 000	480 000	$S \rightarrow S: 100$
2	30 000	477 400	$S \rightarrow S: 74; S \rightarrow M: 26$
3	20 000	475 000	$S \rightarrow S: 50; S \rightarrow M: 50$
4	10 000	472 400	$S \rightarrow S: 24; S \rightarrow M: 76$
5	0	470 000	$S \rightarrow M: 100$

Fonte: próprio autor (2026).

passos de menor h poderiam tornar-se *infectíveis* e seriam descartados. Este é exatamente o cálculo que o sistema executa para cada otimização e exibe como o gráfico de dispersão da fronteira eficiente.

3.5 Metodologia de desenvolvimento

Para que um sistema seja construído de forma escalável e de fácil manutenção, faz-se necessário utilizar padrões de projeto consolidados, que tornam o código limpo, concreto e estruturado. Martin (2000) descreve alguns destes padrões que serão utilizados neste trabalho.

A arquitetura do *backend* foi organizada segundo os princípios do *Domain-Driven Design* (DDD), abordagem proposta por Evans (2003) que orienta o desenvolvimento de *software* a partir do modelo de domínio do problema. O código é estruturado em quatro camadas com responsabilidades bem delimitadas:

- **Domínio** (`CropRotation.Domain`): contém as entidades, objetos de valor, raízes de agregado, eventos de domínio e interfaces de repositório. Esta camada não possui dependências externas e representa o núcleo do sistema;
- **Aplicação** (`CropRotation.Application`): orquestra os casos de uso por meio de serviços de aplicação, objetos de transferência de dados (DTOs), comandos e validadores. Depende apenas da camada de domínio;
- **Infraestrutura** (`CropRotation.Infrastructure`): implementa as interfaces definidas no domínio, fornecendo repositórios concretos (EF Core), o serviço de autenticação JWT, o serviço de importação de dados via planilha e o serviço solucionador de otimização (OR-Tools);
- **API** (`CropRotation.Api`): expõe os casos de uso como *endpoints* REST, configurando roteamento, autenticação e documentação via Swagger.

Dentro da camada de domínio, as entidades principais são modeladas como raízes de agregado (*Aggregate Roots*): `Crop`, `Farm`, `OptimizationRun` e `User`. Cada raiz de agregado encapsula as invariantes do domínio e emite eventos de domínio ao final de operações significativas — por exemplo, `OptimizationCompletedEvent` é disparado após a conclusão de uma execução de otimização.

A comunicação entre *frontend* e *backend* segue o estilo arquitetural REST (*Representational State Transfer*), com transferência de dados no formato JSON (JavaScript Object Notation). Os *endpoints* são agrupados por recurso sendo eles: `/crops`; `/farms`; `/optimizationruns` e `/auth`.

A autenticação é implementada por meio de tokens JWT (*JSON Web Token*) (JONES; BRADLEY; SAKIMURA, 2015): ao efetuar *login*, o cliente recebe um token

assinado com uma chave secreta configurável; nas requisições subsequentes, o token é enviado no cabeçalho **Authorization: Bearer** e validado pelo *middleware* de autenticação do ASP.NET Core. O *frontend* gerencia o ciclo de vida do token por meio de um interceptador HTTP Angular que insere automaticamente o cabeçalho de autorização em todas as requisições autenticadas.

A validação dos dados de entrada é realizada pelo *framework* FluentValidation (SKINNER, 2024), que permite definir regras de validação em classes dedicadas, separadas dos controladores e serviços. Os erros de domínio são representados por exceções tipadas (`DomainException`) e capturados por um *middleware* centralizado de tratamento de exceções (`ExceptionHandlerMiddleware`), que os converte em respostas HTTP com código de status e mensagem adequados.

4 DESENVOLVIMENTO

Este capítulo descreve o processo de construção do sistema, desde a coleta e organização dos dados históricos necessários ao modelo de otimização até a implementação das camadas de domínio, aplicação, infraestrutura e interface gráfica.

4.1 Pesquisa e preparação dos dados

A aplicação do modelo Target MOTAD requer, para cada cultura participante do planejamento de rotação, três séries históricas anuais: produtividade média (kg/ha), preço médio recebido pelo produtor (R\$/kg) e custo (R\$/ha). Adicionalmente, o modelo incorpora fatores de ajuste de rendimento que refletem o efeito agrônomo de uma cultura antecessora sobre a produtividade da cultura sucessora. Toda essa base de dados foi levantada para o estado de Goiás e abrange dez safras consecutivas (2015/16 a 2024/25).

A Companhia Nacional de Abastecimento (CONAB) possui diversos dados sobre as safras brasileiras, a partir do portal aberto, foi coletado os dados necessários sobre produtividade, preço médio e custos, todos esses dados estão descritos respectivamente nas tabelas 3, 4 e 5.

Foram selecionadas três culturas representativas dos sistemas de produção do Cerrado goiano: soja (*Glycine max* L.), milho (*Zea mays* L.) e algodão (*Gossypium hirsutum* L.). A inclusão do algodão justifica-se pela expressividade da cultura no oeste goiano e pelos acentuados benefícios que ela proporciona quando cultivada em rotação. Todas essas culturas são amplamente cultivadas no estado de Goiás.

Tabela 3 – Produtividade média das culturas — Goiás, safras 2015/16 a 2024/25 (kg/ha).

Safra	Algodão	Soja	Milho
2015/16	1.182	3.120	6.599
2016/17	1.598	3.389	4.228
2017/18	1.615	3.609	6.342
2018/19	1.615	3.480	5.615
2019/20	1.692	3.712	6.853
2020/21	1.694	3.672	6.600
2021/22	1.773	4.025	4.585
2022/23	1.798	3.942	5.076
2023/24	1.994	3.487	6.640
2024/25	1.781	4.183	6.509

CONAB (2025c).

A literatura agrônoma demonstra que o cultivo de uma determinada cultura em sucessão a outra pode elevar ou reduzir o rendimento em relação ao monocultivo, fenômeno capturado no modelo por meio dos coeficientes de retorno líquido ajustados. Com base em boletins técnicos da Embrapa Cerrados e Embrapa Algodão, foram definidos fatores multiplicadores de produtividade para cada par antecessora-sucessora. Esses fatores foram aplicados diretamente nas variáveis de retorno líquido calculadas pela classe

Tabela 4 – Preço médio recebido pelo produtor — Goiás, safras 2015/16 a 2024/25 (R\$/kg).

Safra	Algodão	Soja	Milho
2015/16	3,8744	1,0041	0,3759
2016/17	4,0775	1,1760	0,6216
2017/18	4,8155	1,0169	0,3966
2018/19	6,3080	1,2061	0,4794
2019/20	5,6814	1,2106	0,5168
2020/21	6,8692	1,8129	0,7971
2021/22	11,4667	2,6306	1,2880
2022/23	14,5393	2,8176	1,2498
2023/24	9,9533	2,1814	0,8780
2024/25	8,8722	1,9780	0,8644

Fonte: CONAB (2025b).

Tabela 5 – Custo médio de produção por hectare — Goiás, safras 2015/16 a 2024/25 (R\$/ha).

Safra	Algodão	Soja	Milho
2015/16	6.169,64	2.406,79	2.462,83
2016/17	6.601,44	2.499,17	2.712,75
2017/18	6.303,26	2.638,02	2.864,15
2018/19	6.931,39	3.214,85	2.911,91
2019/20	7.113,22	3.207,26	3.128,48
2020/21	7.534,98	3.211,95	3.366,68
2021/22	7.369,83	3.670,74	3.663,63
2022/23	12.442,55	6.115,76	6.564,63
2023/24	11.707,35	5.117,96	6.477,65
2024/25	13.096,95	4.883,88	5.686,25

Fonte: CONAB (2025a).

`NetReturnCalculatorService`: o rendimento da cultura sucessora é multiplicado pelo fator correspondente.

Para o algodão cultivado em rotação, por exemplo, o modelo aplica um fator de produtividade de 1,12 quando a antecessora é a soja — valor consistente com a literatura que registra ganhos de 10 a 15% no rendimento do algodão em sistemas de rotação bem conduzidos (Embrapa, 2021). Para o milho sucedendo a soja, aplica-se fator de 1,09 sobre o rendimento.

4.2 Modelagem do domínio

O domínio do sistema foi modelado seguindo os padrões do DDD, com ênfase na expressividade do código em relação aos conceitos do negócio. Foram identificadas quatro raízes de agregado principais:

Crop (cultura agrícola) encapsula os dados cadastrais da cultura, sua série histórica de produção (`CropHistoricalData`) e os fatores de ajuste de rendimento por rotação

(`RotationYieldAdjustment`). A raiz é responsável por validar a unicidade dos registros históricos por período e UF, bem como por preparar a série histórica para o solver por meio do método `PrepareHistoryForOptimization`, que agrega os registros mensais em médias anuais filtradas por estado.

Farm (propriedade rural) armazena o nome, a área total (ha), a unidade federativa e a identificação do usuário proprietário. A UF da propriedade determina quais registros históricos são selecionados para a otimização, garantindo que o planejamento seja calibrado com os preços e produtividades da região onde a fazenda está localizada.

OptimizationRun (execução de otimização) é a raiz de agregado mais complexa do sistema. Ela registra os parâmetros de entrada de uma execução do modelo — nome, propriedade, renda-alvo T , culturas selecionadas, número de passos de risco e restrições laterais — gerencia o ciclo de vida da execução (estados *Pending*, *Running*, *Completed*, *Failed* e *InsufficientData*) e armazena a coleção de soluções geradas na fronteira eficiente (`OptimizationSolution`). Ao concluir com sucesso, emite o evento de domínio `OptimizationCompletedEvent`.

User (usuário) gerencia as credenciais, o papel (*Admin* ou *Common*) e a autenticação baseada em senha com *hash* BCrypt.

Além das raízes de agregado, foram definidos objetos de valor (*Value Objects*) que encapsulam conceitos sem identidade própria. O objeto `RotationKey` identifica um par de rotação ($i \rightarrow j$), implementando igualdade por valor e oferecendo operações como `Reverse()` e a propriedade `IsMonoculture`. A entidade `RotationConstraint` representa as restrições laterais do modelo (Equações 6 e 7), tipificadas pelo enumerador `ConstraintType` (*MinAreaPerRotation*, *MaxAreaPerRotation*, *MinTotalAreaPerCrop*, *MaxTotalAreaPerCrop* e *EqualAreaDivision*).

4.3 Implementação das camadas de aplicação e infraestrutura

Com os dados preparados e o domínio modelado, as camadas de aplicação e infraestrutura foram implementadas para orquestrar os casos de uso do sistema e conectar o modelo de domínio às tecnologias externas. Esta seção descreve os serviços de aplicação responsáveis pela execução das otimizações, a integração com o solver OR-Tools e as configurações de mapeamento objeto-relacional da camada de persistência.

A camada de aplicação orquestra os casos de uso expondo serviços que combinam repositórios, serviços de domínio e a interface do solver. O `OptimizationService` é o serviço central do sistema: ao receber uma requisição de execução (`RunAsync`), ele recupera a `OptimizationRun` e a `Farm` associada, carrega as culturas com seus dados históricos, filtra os registros pelo estado da propriedade, calcula os retornos líquidos médios e históricos por meio do `NetReturnCalculatorService` e monta o objeto `SolverRequest` passado ao solver. Ao receber as soluções, persiste cada ponto da fronteira eficiente como um `OptimizationSolution` com suas alocações de área detalhadas.

O `NetReturnCalculatorService` implementa o cálculo dos coeficientes RL_{ij} do modelo. Para cada par de rotação ($i \rightarrow j$), o retorno líquido é calculado como:

$$RL_{ij} = P_j \cdot Y_j \cdot f_{ij} - (C_j - e_{ij} \cdot p_N) \quad (1)$$

em que P_j é o preço médio da cultura j , Y_j sua produtividade média, f_{ij} o fator de ajuste de rendimento quando j sucede i , C_j o custo variável de j , e_{ij} a economia de nitrogênio (kg/ha) e p_N o preço unitário do nitrogênio (R\$/kg). Para o cálculo dos coeficientes históricos, o mesmo procedimento é aplicado a cada ano k da série histórica.

O `CropRotationSolverService` implementa a interface `ICropRotationSolver` e recebe um `SolverRequest` contendo as culturas, os coeficientes de retorno líquido médio e histórico, os parâmetros do modelo e as restrições laterais. O serviço gera a fronteira eficiente parametrizando o nível de risco h de h_{\max} até 0 em n passos equiespaçados.

Para cada valor de h , um problema de programação linear independente é instanciado e resolvido pelo GLOP:

1. Criação das variáveis de decisão $x_{ij} \geq 0$ para cada par de rotação (Equação 1);
2. Criação das variáveis auxiliares de desvio $Y_k \geq 0$ para cada ano histórico k ;
3. Definição da função objetivo de maximização da receita total (Equação 1);
4. Adição da restrição de oferta $\sum_{i,j} x_{ij} = TFA$ (Equação 2);
5. Adição das restrições de transbordo $x_{ij} = x_{ji}$ para todo par $i \neq j$ (Equações 3/5);
6. Adição das restrições de risco $\sum_{ij} RL_{ijk} \cdot x_{ij} + Y_k \geq T$ para cada ano k (Equação 8);
7. Adição da restrição de limite de risco $\frac{1}{m} \sum_k Y_k \leq h$ (Equação 9);
8. Adição das restrições laterais configuradas pelo usuário (Equações 6 e 7).

O valor h_{\max} é determinado resolvendo o problema sem a restrição de limite de risco, obtendo o nível de risco do ótimo irrestrito. As soluções infactíveis (para valores de h muito baixos) são descartadas, gerando apenas os pontos Pareto-ótimos da fronteira eficiente.

O mapeamento objeto-relacional foi configurado por meio de classes `IEntityTypeConfiguration<T>` para cada agregado. As configurações definem chaves primárias, colunas com tipos e precisão adequados (por exemplo, `decimal(18,4)` para valores monetários e de área), relacionamentos com cascade delete onde apropriado e conversores de enum para string no banco de dados.

4.4 Interface de programação de aplicação (API REST)

A API expõe quatro grupos de *endpoints* versionados sob o prefixo `/api/v1/`: `/crops`, `/farms`, `/optimizationruns` e `/auth`. Todos os *endpoints*, exceto o de autenticação, exigem um token JWT válido no cabeçalho `Authorization`.

O controle de acesso é realizado em dois níveis. Os *endpoints* de leitura e criação de fazendas e otimizações são acessíveis a qualquer usuário autenticado, mas retornam apenas os recursos pertencentes ao usuário autenticado. Os *endpoints* de gestão de culturas (cadastro, importação, histórico e exclusão) estão restritos ao papel *Admin*, pois as culturas e seus dados históricos constituem uma base compartilhada por todos os usuários do sistema.

O fluxo de execução de uma otimização via API ocorre em dois passos: (1) criação da `OptimizationRun` com os parâmetros desejados via `POST /optimizationruns`, que retorna o ID do recurso criado; (2) disparo da execução via `POST /optimizationruns/{id}/run`, que executa o solver de forma síncrona e retorna a fronteira eficiente completa. O *endpoint* `GET /optimizationruns/{id}/frontier` permite recuperar a fronteira de uma execução anterior sem re-executar o solver.

A documentação interativa da API é gerada automaticamente pelo Swashbuckle e disponibilizada em `/swagger` no ambiente de desenvolvimento, permitindo testar os *endpoints* diretamente pelo navegador com autenticação JWT.

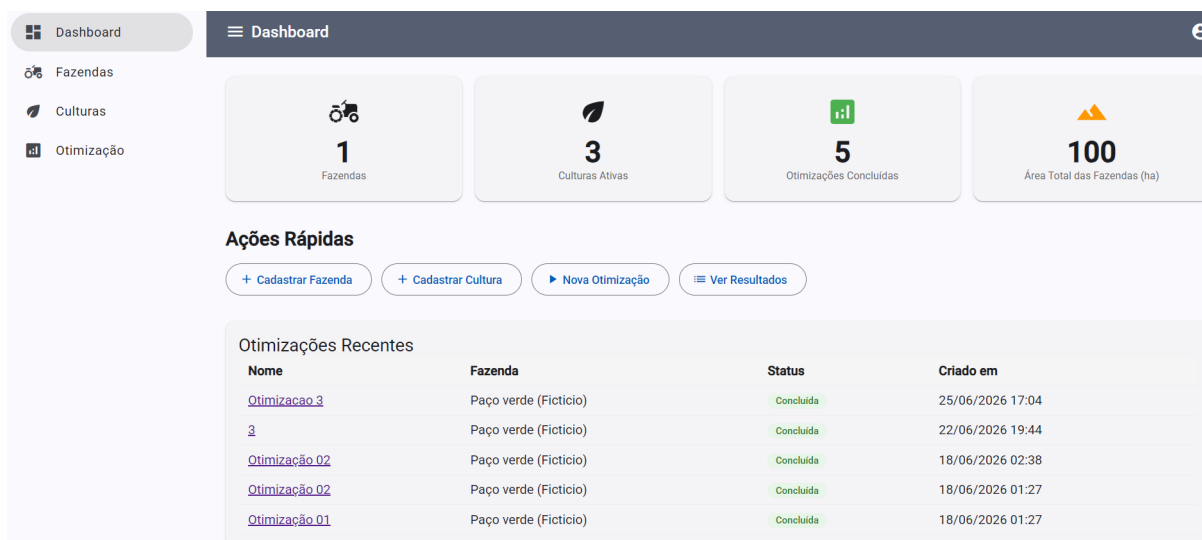
4.5 Interface web

A interface web é o ponto de contato entre o sistema e o produtor rural, devendo oferecer uma experiência de uso clara e orientada ao fluxo de trabalho de planejamento de rotações. Desenvolvida em Angular 21 com componentes do Angular Material, a aplicação é organizada em telas funcionais acessíveis por rotas protegidas por autenticação. Esta seção descreve a estrutura de navegação e as telas que compõem a interface.

A aplicação Angular é composta por seis telas principais, acessadas por rotas protegidas por *guards* de autenticação e, no caso das telas administrativas, por *guard* de papel. O componente de *layout* provê a navegação lateral (*sidenav*) com atalhos para todas as seções da aplicação.

A tela inicial (*dashboard*) demonstrada na Figura 5, apresenta um resumo da conta do usuário: número de fazendas cadastradas, total de culturas disponíveis, quantidade de otimizações concluídas e lista das cinco execuções mais recentes com seus respectivos *status*.

Figura 5 – Tela inicial — resumo de fazendas, culturas e execuções recentes.



Fonte: próprio autor (2026).

A tela de fazendas permite cadastrar, editar e excluir propriedades rurais. Cada fazenda é identificada por nome, área total (ha), estado (UF) e localização opcional. A tela de culturas, restrita ao perfil *Admin*, lista as culturas cadastradas e permite importar novos registros via planilha XLSX, com exibição de relatório de erros e importações bem-sucedidas ao final do processo.

O assistente de criação de otimização foi implementado como um componente *stepper* de múltiplas etapas, guiando o usuário por quatro passos: (1) definição dos parâmetros gerais (nome, fazenda, renda-alvo T , número de pontos na fronteira), ilustrado na Figura 6; (2) seleção das culturas a incluir no planejamento, ilustrado na Figura 7; (3) parâmetro opcional para excluir opções de monocultura e definição da área mínima para cada cultura selecionada, ilustrado na Figura 8; (4) revisão da configuração completa e confirmação de execução, ilustrado na Figura 9. Ao concluir o formulário, o *frontend* cria a execução via API e, imediatamente, dispara o cálculo do solver, redirecionando o usuário à tela de resultados ao término.

Figura 6 – Assistente de otimização — Etapa 1.

Fonte: próprio autor (2026).

Figura 7 – Assistente de otimização — Etapa 2.

Fonte: próprio autor (2026).

A tela de resultados representada na Figura 10 é a peça central da interface do sistema. Ela apresenta a fronteira eficiente como um gráfico de dispersão (Chart.js), com o eixo horizontal representando a medida de risco h e o eixo vertical a receita total esperada (R\$). Cada ponto do gráfico corresponde a uma solução Pareto-ótima; ao clicar em um ponto, o usuário visualiza os detalhes daquela solução: tabela de alocações de área por par de rotação, retorno líquido por hectare e receita total, além de um gráfico de rosca (*doughnut*) que ilustra a distribuição proporcional de área entre as rotações planejadas.

A tela exibe também uma tabela com todas as soluções da fronteira conforme Figura 11, permitindo ao produtor comparar sistematicamente as opções de planejamento e selecionar a que melhor equilibra retorno esperado e tolerância ao risco, de acordo com sua realidade financeira e preferências de gestão.

Figura 8 – Assistente de otimização — Etapa 3.

Dashboard

Fazendas

Culturas

Otimização

← Nova Otimização

1 Configuração — 2 Culturas — 3 Restrições — 4 Execução

Configure restrições opcionais para o modelo de rotação.

Permitir monocultura – quando desmarcado, força rotação entre culturas diferentes

Área mínima por cultura (opcional)
 Defina a área mínima alocada a cada cultura. Deixe em branco para sem restrição.

Cultura	Área mínima
Algodão CT	– ha
Milho CR	– ha
Soja SB	– ha

< Voltar > Próximo

Fonte: próprio autor (2026).

Figura 9 – Assistente de otimização — Etapa 4.

Dashboard

Fazendas

Culturas

Otimização

← Nova Otimização

1 Configuração — 2 Culturas — 3 Restrições — 4 Execução

Revise os dados abaixo antes de iniciar a otimização.

CONFIGURAÇÃO

Nome	Otimização 02
Fazenda	Paço verde (Fictício) – 100 ha
Receita Bienal Alvo	R\$ 200.000
Passos de Risco	20

CULTURAS SELECIONADAS

Algodão	CT
Milho	CR
Soja	SB

RESTRIÇÕES

Permitir monocultura	Não
----------------------	-----

< Voltar > Criar & Executar

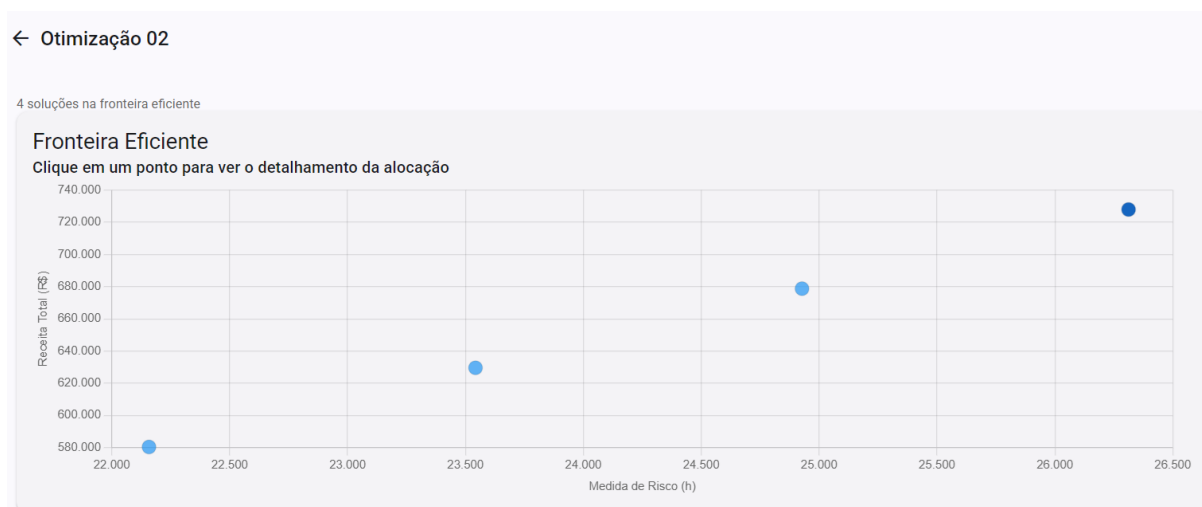
Fonte: próprio autor (2026).

4.6 Cálculo dos retornos líquidos por par de rotação

Uma das etapas centrais da construção do sistema foi a implementação do serviço `NetReturnCalculatorService`, responsável por transformar os dados históricos brutos nos coeficientes de retorno líquido utilizados pelo modelo de otimização. O serviço opera em duas fases sequenciais para cada par de rotação ($i \rightarrow j$): na primeira fase, computa os retornos líquidos históricos anuais $RL_{ij,k}$; na segunda, calcula as médias simples \overline{RL}_{ij} sobre o período histórico.

Para cada combinação de cultura antecessora c_i e cultura sucessora c_j , o serviço

Figura 10 – Resultados — Gráfico de dispersão da fronteira eficiente.



Fonte: próprio autor (2026).

Figura 11 – Resultados — Fronteira eficiente e Culturas em rotação.



Fonte: próprio autor (2026).

busca o fator de ajuste de rendimento aplicável ao par. A busca segue uma hierarquia de prioridade: primeiramente verifica-se a existência de um ajuste específico registrado para o par ($c_i \rightarrow c_j$); caso não exista, aplica-se o ajuste genérico da cultura c_j (aquele com antecessora indefinida), que captura o bônus médio de rotação independente da cultura antecessora. Se nenhum ajuste for encontrado, o retorno é calculado sem modificação, representando o desempenho da cultura em monocultura.

Quando um fator de ajuste está presente, o cálculo modifica a produtividade da cultura sucessora utilizando o fator $f_{ij} \geq 1$. O retorno líquido ajustado para o par ($i \rightarrow j$) no ano k é, portanto:

$$RL_{ij,k} = P_{j,k} \cdot Y_{j,k} \cdot f_{ij} - C_j \quad (1)$$

onde $P_{j,k}$ é o preço médio recebido pelo produtor no ano k , $Y_{j,k}$ a produtividade histórica e C_j o custo variável de referência da cultura j . O retorno médio \overline{RL}_{ij} é obtido pela média simples de $RL_{ij,k}$ sobre todos os anos k com dados disponíveis para a UF da fazenda.

O conjunto de resultados produzido pelo serviço — o dicionário de médias \overline{RL} e o dicionário histórico RL_{hist} indexado por (i, j, k) — constitui a entrada direta para a construção do modelo de programação linear. A correta aplicação dos fatores agrônômicos nessa etapa é determinante para que o solver capture os benefícios econômicos da diversificação botânica já no nível dos coeficientes do modelo.

4.7 Construção e resolução do modelo de programação linear

A resolução do modelo Target MOTAD foi implementada no `CropRotationSolverService`, que recebe um objeto `SolverRequest` contendo os coeficientes calculados na etapa anterior, os parâmetros do modelo e as restrições laterais configuradas pelo usuário. O serviço gera a fronteira eficiente por meio de uma abordagem paramétrica: o nível máximo admissível de risco h é variado de h_{max} até zero em n passos equiespaçados, e para cada valor de h um problema de programação linear independente é construído e resolvido pelo solver GLOP do Google OR-Tools.

O primeiro passo da execução é a determinação de h_{max} , realizada resolvendo o modelo sem a restrição de limite de risco (Equação 9). A solução ótima desse problema irrestrito fornece os valores das variáveis auxiliares Y_k^* cujo desvio médio absoluto, $h_{\text{max}} = \frac{1}{m} \sum_k Y_k^*$, define o extremo de maior retorno (e maior risco) da fronteira eficiente. A partir desse valor, o vetor de níveis de risco $\mathbf{h} = [h_{\text{max}}, \dots, 0]$ é gerado com n pontos equidistantes.

Para cada valor $h_s \in \mathbf{h}$, o serviço instancia um novo objeto do solver através do método da biblioteca `Solver.CreateSolver("GLOP")` e constrói o problema linear em etapas bem definidas. Primeiro, são criadas as n^2 variáveis de decisão $x_{ij} \geq 0$ — uma para cada par de rotação — e as m variáveis auxiliares de risco $Y_k \geq 0$, uma por ano histórico. A função objetivo é configurada para maximização, com os coeficientes \overline{RL}_{ij} associados a cada variável x_{ij} .

Em seguida, são adicionadas as restrições estruturais do modelo: a restrição de oferta impõe que a soma de todas as áreas alocadas seja exatamente igual à área total da fazenda (TFA); as restrições de transbordo garantem que a área destinada ao par $(i \rightarrow j)$ seja igual à área do par inverso $(j \rightarrow i)$ para todo $i \neq j$, assegurando a divisão simétrica da propriedade entre os dois anos do ciclo bianual. Para cada ano histórico k , uma restrição de risco define implicitamente $Y_k = \max(0, T - \text{renda}_k)$, ou seja, o desvio da renda realizada em relação à renda-alvo T . Por fim, a restrição de limite de risco impõe que o desvio médio absoluto não exceda h_s .

As restrições laterais configuradas pelo usuário são aplicadas após as restrições estruturais. Cada objeto `RotationConstraint` ativo é traduzido em uma ou mais restrições no solver: limites mínimo e máximo de área por par de rotação, limites de área total por cultura e a proibição de monocultura ($x_{ii} = 0$). Ao concluir a construção, `solver.Solve()` é invocado e o status de retorno é verificado: apenas soluções com status `OPTIMAL` são aceitas e registradas como pontos da fronteira eficiente; os demais casos — tipicamente problemas infactíveis para valores muito restritivos de h — são descartados sem interromper o processo.

4.8 Integração entre as camadas e fluxo de execução

O fluxo completo de uma execução de otimização atravessa todas as camadas do sistema em uma sequência orquestrada pelo `OptimizationService`. O processo se inicia com uma chamada `POST /api/v1/optimizationruns`, que persiste a configuração de execução com status *Pending* e retorna o identificador do recurso criado. Imediatamente após, o *frontend* aciona `POST /api/v1/optimizationruns/{id}/run`, que delega ao serviço de aplicação a responsabilidade pela preparação dos dados e pelo disparo do solver.

Na camada de aplicação, o serviço recupera a `OptimizationRun`, a fazenda associada e todas as culturas selecionadas com seus respectivos dados históricos. A validação de dados mínimos verifica se ao menos duas culturas possuem registros históricos para a UF da fazenda; caso contrário, a execução é marcada como *InsufficientData* e uma exceção de domínio é lançada. Para as culturas válidas, o método `PrepareHistoryForOptimization` filtra os registros pela UF e agrega os dados por ano safra, produzindo as séries históricas prontas para o cálculo dos coeficientes.

Com os coeficientes calculados e o `SolverRequest` montado, a execução é marcada como *Running* e o controle é transferido ao `CropRotationSolverService`. O solver retorna uma lista de pontos da fronteira eficiente, cada um contendo o nível de risco h_s , a receita total esperada Z_s e os valores das variáveis de decisão x_{ij} . O serviço de aplicação filtra os pontos com status ótimo, converte cada um em uma entidade `OptimizationSolution` com suas respectivas `SolutionAllocation` — apenas as alocações com área superior a 0,01 ha são registradas —, e avança a execução para o estado *Completed*, emitindo o evento de domínio `OptimizationCompletedEvent`.

A persistência da `OptimizationRun` completa com suas soluções e alocações é realizada em uma única operação pelo repositório EF Core, que persiste o grafo de entidades em cascata. A API retorna então o `EfficientFrontierDto`, contendo todos os pares (h_s, Z_s) da fronteira eficiente, que o componente Angular utiliza para renderizar o gráfico de dispersão e popular a tabela comparativa de soluções. A máquina de estados da `OptimizationRun` — *Pending*, *Running*, *Completed*, *Failed* e *InsufficientData* — garante rastreabilidade do ciclo de vida de cada execução e permite ao usuário recuperar os resultados de execuções anteriores a qualquer momento por meio do *endpoint* `GET /optimizationruns/{id}/frontier`, sem necessidade de re-executar o solver.

4.9 Algoritmo de Otimização

O algoritmo 1 resolve um único ponto da **fronteira eficiente risco-renda** para o modelo *Target MOTAD* de rotação de culturas. É invocado repetidamente pelo método SOLVE com valores decrescentes de h (de h_{\max} até 0), gerando uma nova solução entre renda esperada e risco.

Papel de cada componente

Variáveis x_{ij} Representam a área (ha) dedicada à sequência “cultura i no primeiro ano \rightarrow cultura j no segundo ano”. São as variáveis de decisão do planejamento: quanto alocar a cada rotação.

Variáveis Y_k Medem o quanto a renda do ano histórico k , aplicada ao plano atual, ficou *abaixo* da meta T . Se a renda supera T , $Y_k = 0$ (sem desvio).

Objetivo (Eq. 1) Maximizar o retorno líquido médio esperado somado sobre todos os pares de rotação ponderados pela área alocada.

Área total (Eq. 2) Toda a área da fazenda TFA deve ser alocada, sem sobra nem falta.

Simetria de transbordo (Eq. 3) A área dedicada à rotação ($i \rightarrow j$) deve ser igual à de ($j \rightarrow i$), modelando uma rotação bienal equilibrada.

Risco por ano (Eq. 4) Para cada ano histórico k , Y_k absorve o deficit de renda em relação à meta T .

Limite de risco (Eq. 5) O desvio médio anual $\frac{1}{m} \sum_k Y_k$ (risco MOTAD) não pode superar h . Quanto menor h , mais conservador o plano.

Restrições laterais Restrições agrônômicas como área mínima por cultura e proibição de monocultura ($x_{ii} = 0$).

Geração da Fronteira Eficiente

O método SOLVE calcula h_{\max} resolvendo o LP sem a restrição de risco (Eq. 5), obtendo o ponto de máxima renda. Em seguida, invoca SOLVEFORRISK para valores $h \in \{h_{\max}, \dots, 0\}$ em passos uniformes, traçando a fronteira eficiente.

4.10 Código fonte

O sistema foi disponibilizado publicamente como código aberto (SOUSA, 2026).

Algoritmo 1: SolveForRisk — Target MOTAD

Entrada: *request* (culturas \mathcal{C} , TFA, T , retornos médios \overline{NR} e históricos NR_{hist} , restrições laterais), $h \geq 0$, *solutionNumber*

Saída: SolverSolutionPoint

- 1 **início**
- 2 *solver* \leftarrow CriarSolver(GLOP);
- 3 **para** cada $i \in \mathcal{C}$ **faça**
- 4 **para** cada $j \in \mathcal{C}$ **faça**
- 5 $x_{ij} \leftarrow$ solver.CriarVariável(0, $+\infty$);
- 6 **fim**
- 7 **fim**
- 8 $\mathcal{K} \leftarrow$ ObterAnosHistóricos(*request*);
- 9 $m \leftarrow |\mathcal{K}|$;
- 10 **para** cada $k \in \mathcal{K}$ **faça**
- 11 $Y_k \leftarrow$ solver.CriarVariável(0, $+\infty$);
- 12 **fim**
- 13 *objetivo* \leftarrow $\max \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} \overline{NR}_{ij} \cdot x_{ij}$;
- 14 solver.DefinirObjetivo(*objetivo*);
- 15 solver.AdicionarRestrição $\left(\sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} x_{ij} = \text{TFA} \right)$;
- 16 **para** cada par $(i, j) \in \mathcal{C} \times \mathcal{C}$ com $i < j$ **faça**
- 17 solver.AdicionarRestrição ($x_{ij} - x_{ji} = 0$);
- 18 **fim**
- 19 **para** cada $k \in \mathcal{K}$ **faça**
- 20 solver.AdicionarRestrição $\left(\sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} NR_{ijk} \cdot x_{ij} + Y_k \geq T \right)$;
- 21 **fim**
- 22 solver.AdicionarRestrição $\left(\frac{1}{m} \sum_{k \in \mathcal{K}} Y_k \leq h \right)$;
- 23 AplicarRestriçõesLaterais(*solver*, x , \mathcal{C} , *request*.Restrições);
- 24 *status* \leftarrow solver.Resolver();
- 25 **se** *status* \neq *OPTIMAL* **então**
- 26 **retorna** { *IsOptimal* \leftarrow **false**, $h \leftarrow h$, *solutionNumber* \leftarrow *solutionNumber* };
- 27 **fim**
- 28 *income* \leftarrow 0;
- 29 **para** cada $(i, j) \in \mathcal{C} \times \mathcal{C}$ **faça**
- 30 $area_{ij} \leftarrow$ x_{ij} .ValorSolução();
- 31 *income* \leftarrow *income* + $area_{ij} \cdot \overline{NR}_{ij}$;
- 32 **fim**
- 33 **retorna** { *IsOptimal* \leftarrow **true**, $h \leftarrow h$, *TotalIncome* \leftarrow *income*, *AreaByRotation* \leftarrow *area* };
- 34 **fim**

5 RESULTADOS

Este capítulo apresenta os resultados obtidos com a aplicação do sistema de otimização de rotação de culturas em um cenário de planejamento construído para fins de validação. Três execuções de otimização foram realizadas sobre a mesma propriedade e base de culturas, variando os parâmetros de restrição. Para cada execução são descritos os parâmetros configurados, a fronteira eficiente gerada pelo modelo e a alocação de área associada a soluções representativas, com o objetivo de ilustrar como o sistema apoia a tomada de decisão do produtor rural.

5.1 Configuração do cenário de teste

As otimizações foram executadas sobre a propriedade fictícia denominada “Pão Verde”, com área total de 100 ha. O uso de uma propriedade fictícia permite validar o comportamento do modelo em condições controladas, sem exposição de dados reais de produtores.

As três culturas selecionadas para compor o planejamento de rotação foram Soja (código SB), Milho (código CR) e Algodão (código CT), culturas representativas dos principais sistemas produtivos do Cerrado goiano. A base histórica utilizada abrange as safras 2015/16 a 2024/25 para o estado de Goiás, em conformidade com os dados levantados no Capítulo 4.

A renda bianual alvo foi definida em $T = R\$ 200.000,00$ para todos os experimentos, o que corresponde a R\$ 1.000,00/ha por ano — patamar conservador que serve como referência para o cômputo dos desvios de risco pelo modelo Target MOTAD (SALASSI; DELIBERTO; GUIDRY, 2013). A Tabela 6 sintetiza as configurações das três execuções analisadas.

Tabela 6 – Configuração das execuções de otimização.

Parâmetro	Exec. 1	Exec. 2	Exec. 3
Nome	Otimização 01	Otimização 02	Otimização 03
Fazenda	Pão Verde	Pão Verde	Pão Verde
Área total (ha)	100	100	100
Culturas	SB, CR, CT	SB, CR, CT	SB, CR, CT
Renda alvo bianual (R\$)	200.000	200.000	200.000
Passos de risco	20	20	10
Restrição sem monocultura	Não	Sim	Sim

Fonte: próprio autor (2026).

5.2 Visão geral da interface de resultados

Antes de detalhar cada execução, convém descrever os elementos da interface gráfica que o sistema apresenta ao usuário ao final de cada otimização, pois os conceitos ali expostos fundamentam a leitura dos resultados das seções seguintes.

Assistente de criação (wizard). A criação de uma nova otimização é guiada por um assistente de quatro etapas. Na primeira etapa o usuário informa o nome da execução, seleciona a fazenda, define a renda bianual alvo T e o número de passos de risco, que determina a resolução da fronteira eficiente gerada. Na segunda etapa são selecionadas as culturas a incluir no modelo (mínimo de duas). Na terceira etapa são configuradas as restrições opcionais: a opção “Permitir monocultura”, quando desmarcada, acrescenta ao modelo a restrição $x_{ii} = 0 \forall i$, que proíbe o plantio da mesma cultura nos dois anos do ciclo bianual; é possível ainda definir área mínima por cultura, que gera restrições do tipo $\sum_j x_{ij} \geq \underline{A}_i$. Na quarta e última etapa o usuário revisa os dados e dispara a execução.

Gráfico da fronteira eficiente. Após a conclusão da otimização, a tela de resultados apresenta um gráfico de dispersão intitulado “Fronteira Eficiente”. Cada ponto representa uma solução Pareto-ótima do modelo: o eixo horizontal exibe o valor de risco h (medida Target MOTAD, em R\$) e o eixo vertical exibe a receita bianual esperada (em R\$). A relação entre os dois eixos é de *tradeoff*: ao mover-se para a direita na curva, o produtor aceita maior risco em troca de maior renda esperada; ao mover-se para a esquerda, reduz o risco ao custo de uma renda menor. O ponto mais à direita corresponde à solução de máxima renda (sem restrição de risco), enquanto o ponto mais à esquerda corresponde à solução de mínimo risco viável dado o conjunto de restrições configurado.

Tabela de soluções. Abaixo do gráfico, a tabela de soluções lista todas as soluções da fronteira com quatro colunas: número sequencial da solução (**#**), receita bianual esperada em R\$ (**Receita**), medida de risco h em R\$ (**Risco h**) e desvio médio absoluto em R\$ (**DMA**). O DMA é igual ao dobro de h , pois representa o desvio médio absoluto total (bilateral), enquanto h captura apenas a parcela de desvio negativo em relação à renda alvo T — propriedade do modelo Target MOTAD (SALASSI; DELIBERTO; GUIDRY, 2013). O usuário pode clicar em qualquer linha para inspecionar o detalhamento de alocação da solução correspondente.

Detalhamento de alocação. Ao selecionar uma solução, um painel lateral exibe a tabela de alocação com as colunas **Rotação** (cultura do Ano 1 \rightarrow cultura do Ano 2), **Área (ha)** (hectares alocados ao par de rotação), **RL/ha** (retorno líquido esperado por hectare do par, em R\$) e **Total** (receita total do par, em R\$). Pares de monocultura são destacados visualmente. A tabela de alocação é complementada por um gráfico de rosca (*doughnut*) que representa graficamente a proporção da área total dedicada a cada par de rotação, facilitando a interpretação visual da estratégia de diversificação sugerida pelo modelo.

5.3 Execução 1 — sem restrição de monocultura

A primeira execução (denominada “Otimização 01”) foi configurada com 20 passos de risco e sem nenhuma restrição lateral adicional, permitindo ao modelo explorar livremente todas as combinações de rotação possíveis, incluindo monocultura.

Fronteira eficiente

O solver gerou 15 soluções viáveis sobre a fronteira eficiente, sendo que os 5 passos de menor risco solicitados ao longo dos 20 pontos parametrizados resultaram em problemas inviáveis, indicando que não existe nenhuma combinação de alocação capaz de atingir valores de risco inferiores a aproximadamente R\$ 6.924,33 com as culturas e dados históricos fornecidos. A Tabela 7 apresenta todas as soluções geradas.

A amplitude de renda observada entre a solução de maior renda (R\$ 727.963,63) e a de menor risco (R\$ 681.675,77) corresponde a uma redução de aproximadamente

Tabela 7 – Fronteira eficiente — Exec. 1 (sem restrição de monocultura).

Sol. #	Receita bianual (R\$)	Risco h (R\$)	DMA (R\$)
1	727.963,63	26.312,44	52.624,89
2	725.700,20	24.927,58	49.855,15
3	723.436,76	23.542,71	47.085,42
4	720.245,26	22.157,85	44.315,69
5	716.738,94	20.772,98	41.545,96
6	713.232,63	19.388,12	38.776,23
7	709.726,31	18.003,25	36.006,50
8	706.219,99	16.618,38	33.236,77
9	702.713,67	15.233,52	30.467,04
10	699.207,36	13.848,65	27.697,31
11	695.701,04	12.463,79	24.927,58
12	692.194,72	11.078,92	22.157,85
13	688.688,40	9.694,06	19.388,12
14	685.182,09	8.309,19	16.618,38
15	681.675,77	6.924,33	13.848,65

Fonte: próprio autor (2026).

6,4%, enquanto o risco h recua de R\$ 26.312,44 para R\$ 6.924,33, uma redução de 73,7%. Esse resultado evidencia que o modelo encontrou uma forma eficiente de comprimir significativamente o risco com perda modesta de receita esperada, o que é característica desejável de uma fronteira eficiente bem comportada.

Alocação de área

A Tabela 8 apresenta a composição de área para todas as soluções da fronteira. O Milho não aparece em nenhuma solução: os retornos líquidos do par Soja→Milho (R\$ 3.892,86/ha) e Milho→Soja (R\$ 3.330,22/ha) são substancialmente inferiores ao do par Soja→Algodão (R\$ 8.464,74/ha) e da própria monocultura Soja→Soja (R\$ 6.780,19/ha), tornando o Milho economicamente não competitivo neste cenário.

RL/ha dos pares: Soja→Soja = R\$ 6.780,19/ha; Soja→Algodão = R\$ 8.464,74/ha; Algodão→Soja = R\$ 6.094,53/ha.

O padrão de alocação revela o mecanismo de redução de risco utilizado pelo modelo na ausência de restrições: conforme o nível-alvo de h diminui de solução para solução, a área dedicada ao par Soja↔Algodão é progressivamente substituída por monocultura Soja→Soja. O Algodão, cultura de alta rentabilidade mas também de elevada variabilidade de preços e produtividade, concentra o risco da carteira; ao reduzi-lo em favor da soja pura, o modelo troca rentabilidade por estabilidade.

Observa-se ainda que, em todas as soluções, as áreas dos pares Soja→Algodão e Algodão→Soja são rigorosamente iguais (50,00 ha para a Sol. 1, 47,73 ha para a Sol. 2, e assim por diante). Esse comportamento é consequência direta da restrição de equilíbrio de área incorporada ao modelo, que garante $x_{ij} = x_{ji}$ para todo par de rotação (SALASSI; DELIBERTO; GUIDRY, 2013), assegurando que a área plantada com Algodão no Ano 2 (precedida por Soja) seja igual à área plantada com Soja no Ano 2 (precedida por Algodão), mantendo assim a consistência do planejamento multissafra.

Tabela 8 – Alocação de área por par de rotação — Exec. 1.

Sol. #	Área alocada (ha)			Total (ha)
	Soja→Soja	Soja→Algodão	Algodão→Soja	
1	0,00	50,00	50,00	100,00
2	4,53	47,73	47,73	100,00
3	9,06	45,47	45,47	100,00
4	15,45	42,27	42,27	100,00
5	22,47	38,76	38,76	100,00
6	29,49	35,25	35,25	100,00
7	36,51	31,74	31,74	100,00
8	43,54	28,23	28,23	100,00
9	50,56	24,72	24,72	100,00
10	57,58	21,21	21,21	100,00
11	64,60	17,70	17,70	100,00
12	71,62	14,19	14,19	100,00
13	78,64	10,68	10,68	100,00
14	85,66	7,17	7,17	100,00
15	92,68	3,66	3,66	100,00

Fonte: próprio autor (2026).

5.4 Execução 2 — com restrição de sem monocultura

A segunda execução manteve os mesmos 20 passos de risco da execução anterior, mas acrescentou a restrição *sem monocultura*: $x_{ii} = 0 \forall i$. Essa restrição é relevante em contextos agrônômicos nos quais o produtor, por razões fitossanitárias ou de política de crédito rural, não pode plantar a mesma cultura em dois anos consecutivos.

Fronteira eficiente

Sob a restrição de sem monocultura, o solver gerou apenas 4 soluções viáveis dentre os 20 pontos solicitados, conforme mostra a Tabela 9.

Tabela 9 – Fronteira eficiente — Exec. 2 (Com restrição de sem monocultura).

Sol. #	Receita bianual (R\$)	Risco h (R\$)	DMA (R\$)
1	727.963,63	26.312,44	52.624,89
2	678.744,59	24.927,58	49.855,15
3	629.525,56	23.542,71	47.085,42
4	580.306,52	22.157,85	44.315,69

Fonte: próprio autor (2026).

A comparação com a Exec. 1 é imediata: o h_{\max} coincide (R\$ 26.312,44), pois a solução de máxima renda — alocação integral a Soja↔Algodão, sem monocultura — é naturalmente viável em ambas as execuções. Contudo, sem a possibilidade de utilizar a monocultura como “amortecedor” de risco, o modelo não consegue satisfazer os alvos de h correspondentes às soluções 5 a 20, todas inferiores a R\$ 20.772,98. O valor de risco mínimo

atingível com este conjunto de culturas e dados históricos passa a ser R\$ 22.157,85 — redução de apenas 15,8% em relação ao risco máximo, contra os 73,7% da execução anterior. Esse resultado quantifica o custo agrônomo-financeiro da proibição de monocultura: a fronteira eficiente encolhe drasticamente, oferecendo ao produtor muito menos margem para gerenciar seu risco.

Alocação de área

A Tabela 10 apresenta a composição de área para as 4 soluções da fronteira.

Tabela 10 – Alocação de área por par de rotação — Exec. 2.

Sol. #	Área alocada (ha)				Total (ha)
	Soja→Milho	Soja→Algodão	Milho→Soja	Algodão→Soja	
1	0,00	50,00	0,00	50,00	100,00
2	6,71	43,29	6,71	43,29	100,00
3	13,42	36,58	13,42	36,58	100,00
4	20,13	29,87	20,13	29,87	100,00

Fonte: próprio autor (2026).

RL/ha dos pares: Soja→Milho = R\$ 3.892,86/ha; Milho→Soja = R\$ 3.330,22/ha; Soja→Algodão = R\$ 8.464,74/ha; Algodão→Soja = R\$ 6.094,53/ha.

O papel do Milho na Exec. 2 é fundamentalmente diferente do que se observou na Exec. 1. Aqui, o Milho passa a ser o único mecanismo de redução de risco disponível: ao substituir área de Soja↔Algodão pela rotação Soja↔Milho, o modelo reduz a exposição à volatilidade do Algodão ao custo de retorno médio menor (R\$ 3.330,22 e R\$ 3.892,86/ha contra R\$ 6.094,53 e R\$ 8.464,74/ha). Esse custo de oportunidade explica a queda mais acentuada de renda entre as soluções: a amplitude de receita vai de R\$ 727.963,63 (Sol. 1) a R\$ 580.306,52 (Sol. 4), uma redução de 20,3% para uma compressão de risco de apenas 15,8%.

5.5 Execução 3 — dados históricos revisados

A terceira execução (também denominada “Otimização 03”) repetiu a configuração da Exec. 2 — restrição de sem monocultura ativa — mas com 10 passos de risco e sobre uma base histórica revisada. A revisão consistiu na atualização dos fatores de ajuste de rendimento por rotação, refletindo informações agrônomicas mais recentes, e na incorporação de séries históricas de preços e custos atualizadas.

Fronteira eficiente

A Tabela 11 apresenta as soluções geradas.

Dois efeitos da atualização dos dados são imediatamente visíveis. Primeiro, o h_{\max} caiu de R\$ 26.312,44 (Exec. 2) para R\$ 23.326,64, indicando que os dados históricos revisados apresentam menor dispersão de retornos — as séries mais recentes tendem a refletir um ambiente de preços e produtividades mais estável para as culturas analisadas. Segundo, a receita máxima subiu de R\$ 727.963,63 para R\$ 777.607,48 (+6,8%), reflexo do

Tabela 11 – Fronteira eficiente — Exec. 3.

Sol. #	Receita bianual (R\$)	Risco h (R\$)	DMA (R\$)
1	777.607,48	23.326,64	46.653,28
2	718.175,29	20.734,79	41.469,58
3	658.743,11	18.142,94	36.285,89
4	599.310,93	15.551,09	31.102,19

Fonte: próprio autor (2026).

aumento do retorno líquido do par Algodão→Soja (de R\$ 6.094,53 para R\$ 7.087,41/ha), que eleva o valor esperado da solução de máxima renda.

Quanto à viabilidade da fronteira: com 10 passos de risco e h_{\max} menor, o tamanho de cada passo é de aproximadamente R\$ 2.592,00. Desse total, apenas 4 pontos são viáveis, sinalizando que o piso de risco atingível sem monocultura permanece consideravelmente acima de zero.

Alocação de área

A Tabela 12 detalha a composição de área para as 4 soluções da fronteira da Exec. 3.

Tabela 12 – Alocação de área por par de rotação — Exec. 3.

Sol. #	Área alocada (ha)				Total (ha)
	Soja→Milho	Soja→Algodão	Milho→Soja	Algodão→Soja	
1	0,00	50,00	0,00	50,00	100,00
2	8,72	41,28	8,72	41,28	100,00
3	17,45	32,55	17,45	32,55	100,00
4	26,17	23,83	26,17	23,83	100,00

Fonte: próprio autor (2026).

RL/ha dos pares: Soja→Milho = R\$ 3.836,60/ha; Milho→Soja = R\$ 4.902,28/ha; Soja→Algodão = R\$ 8.464,74/ha; Algodão→Soja = R\$ 7.087,41/ha.

A estrutura qualitativa da alocação é idêntica à da Exec. 2 — a redução de risco é obtida transferindo área de Soja↔Algodão para Soja↔Milho —, porém a magnitude da transferência é diferente. Com Milho→Soja passando de R\$ 3.330,22 para R\$ 4.902,28/ha (+47,2%), o custo de oportunidade de usar o Milho como amortecedor de risco diminui, de modo que o modelo transfere área mais agressivamente: na Sol. 4, são 26,17 ha em Soja↔Milho contra 20,13 ha na Sol. 4 da Exec. 2. Esse comportamento demonstra que a precisão dos fatores de ajuste de rendimento por rotação tem impacto direto tanto no nível de renda da fronteira quanto na estratégia de diversificação recomendada pelo modelo.

5.6 Análise comparativa das execuções

A Tabela 13 consolida as métricas principais das três execuções, permitindo uma análise direta do efeito de cada variável de configuração sobre a fronteira eficiente.

Tabela 13 – Comparativo das execuções de otimização.

Métrica	Exec. 1	Exec. 2	Exec. 3
Soluções viáveis	15	4	4
Receita máxima (R\$)	727.963,63	727.963,63	777.607,48
Receita mínima (R\$)	681.675,77	580.306,52	599.310,93
h_{\max} (R\$)	26.312,44	26.312,44	23.326,64
h_{\min} viável (R\$)	6.924,33	22.157,85	15.551,09
Redução de h atingível (%)	73,7	15,8	33,3
Amplitude de renda (%)	6,4	20,3	22,9
Culturas ativas nas soluções	SB, CT	SB, CR, CT	SB, CR, CT

Fonte: próprio autor (2026).

Efeito da restrição de sem monocultura (Exec. 1 vs. Exec. 2). A introdução da restrição $x_{ii} = 0$ reduziu o número de soluções viáveis de 15 para 4 e limitou a redução de risco atingível de 73,7% para apenas 15,8%. A monocultura Soja→Soja, proibida na Exec. 2, era o principal mecanismo de compressão de risco na Exec. 1 por combinar retorno médio razoável (R\$ 6.780,19/ha) com menor variabilidade histórica em relação ao Algodão. Sem esse recurso, o Milho assume o papel de diversificador, porém com retornos médios mais baixos, gerando custo de renda mais elevado por unidade de risco eliminada.

Efeito da atualização dos dados (Exec. 2 vs. Exec. 3). A revisão da base histórica elevou a receita máxima em 6,8%, reduziu o h_{\max} em 11,4% e ampliou a redução de risco atingível de 15,8% para 33,3%. O aumento do retorno líquido do par Milho→Soja (+47,2%) foi o fator de maior impacto: ao tornar o Milho relativamente mais rentável, os dados revisados ampliam a fronteira eficiente e permitem que o modelo alcance menores níveis de risco sem sacrifício proporcional de renda.

Esses resultados confirmam duas conclusões relevantes para o uso prático do sistema. Em primeiro lugar, a qualidade da base histórica é determinante para a confiabilidade das recomendações do modelo: erros ou desatualizações nos fatores de ajuste de rendimento por rotação distorcem tanto a renda esperada quanto a composição ótima da carteira de culturas. Em segundo lugar, a proibição de monocultura — restrição com fundamento agrônômico sólido — tem consequências financeiras quantificáveis que o sistema expõe diretamente ao produtor, auxiliando a decisão sobre adotar ou não essa prática de manejo.

6 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um sistema web para auxílio ao planejamento de rotações de culturas agrícolas, voltado para propriedades do Cerrado goiano. O sistema implementa o modelo matemático *Target MOTAD* proposto por Salassi, Deliberto e Guidry (2013), permitindo ao produtor rural obter, de forma interativa, a fronteira eficiente de risco-retorno para o planejamento bianual de rotações entre três culturas representativas da região: soja, milho e algodão.

A base de dados histórica utilizada pelo modelo abrange dez safras consecutivas (2015/16 a 2024/25) para o estado de Goiás. Os dados de produtividade foram obtidos das séries históricas de safras da Companhia Nacional de Abastecimento (CONAB) (CONAB, 2025c), enquanto os dados de preços e custos de produção foram extraídos das planilhas de custos de produção disponibilizadas no portal da CONAB (CONAB, 2025a). Os fatores de ajuste de rendimento por rotação foram fundamentados em boletins técnicos da Embrapa (Embrapa, 2021), refletindo os efeitos agronômicos das sequências antecessora-sucessora sobre a produtividade das culturas.

Do ponto de vista técnico, o sistema foi estruturado segundo os princípios do *Domain-Driven Design* (DDD), com separação em quatro camadas bem delimitadas: domínio, aplicação, infraestrutura e API. A resolução do modelo de programação linear foi delegada ao solver GLOP do Google OR-Tools (Google LLC, 2023), que demonstrou eficiência adequada para o porte do problema — até 19 variáveis de decisão e 15 restrições estruturais para três culturas e dez anos históricos. A interface web, desenvolvida com Angular 21 e Angular Material, orienta o produtor por um fluxo de uso intuitivo, desde o cadastro da propriedade até a visualização da fronteira eficiente e a seleção do plano de rotação mais adequado ao seu perfil de risco.

Entre as principais contribuições deste trabalho destacam-se: (i) a adaptação e implementação do modelo *Target MOTAD* proposto por Salassi, Deliberto e Guidry (2013) de rotação de culturas para o contexto do Cerrado goiano, com dados reais de produtividade, preços e custos; (ii) o desenvolvimento de uma ferramenta web acessível que torna o modelo computacionalmente disponível ao produtor sem exigir conhecimento técnico em programação ou otimização; e (iii) a integração de fatores agronômicos de ajuste de rendimento por rotação diretamente nos coeficientes do modelo, capturando os benefícios econômicos da diversificação botânica.

Como limitações, o modelo adotado considera um horizonte de planejamento de dois anos e não incorpora variações sazonais de preços ao longo do ano ou restrições de janelas de plantio específicas de cada cultura.

6.1 Trabalhos futuros

Como perspectivas de trabalho futuro, identificam-se as seguintes extensões para aplicação em cenários reais: (i) ampliação do horizonte de planejamento para três ou mais anos de rotação, pois apenas dois anos não refletem a realidade do agronegócio; (ii) incorporação de dados climáticos e de risco de seca no modelo de risco; (iii) suporte a múltiplos estados e regiões agroclimáticas; (iv) integração com APIs de preços de mercado em tempo real para atualização automática das séries históricas; e (v) inclusão de culturas de cobertura e sistemas de integração lavoura-pecuária no conjunto de espécies disponíveis para rotação.

Referências

- BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. *Linear programming and network flows*. [S.l.]: John Wiley & Sons, 2011. Citado 2 vezes nas páginas 6 e 7.
- CASTELLAZZI, M. et al. A systematic representation of crop rotations. *Agricultural Systems*, Elsevier, v. 97, n. 1-2, p. 26–33, 2008. Citado na página 2.
- CONAB. *Custos de Produção por Produto e Estado*. 2025. Companhia Nacional de Abastecimento — CONAB. Acesso em: 06 jun. 2026. Disponível em: <<https://www.conab.gov.br/info-agro/custos-de-producao>>. Citado 2 vezes nas páginas 19 e 37.
- CONAB. *Preços Agropecuários — Séries Históricas por Estado*. 2025. Companhia Nacional de Abastecimento — CONAB. Acesso em: 06 jun. 2026. Disponível em: <<https://www.conab.gov.br/info-agro/precos>>. Citado na página 19.
- CONAB. *Séries Históricas de Produtividade por Estado*. 2025. Companhia Nacional de Abastecimento — CONAB. Acesso em: 06 jun. 2026. Disponível em: <<https://www.conab.gov.br/info-agro/safras/serie-historica-das-safras>>. Citado 2 vezes nas páginas 18 e 37.
- DANTZIG, G. B. Linear programming. *Operations research*, INFORMS, v. 50, n. 1, p. 42–47, 2002. Citado na página 6.
- DETLEFSEN, N. K.; JENSEN, A. L. Modelling optimal crop sequences using network flows. *Agricultural Systems*, Elsevier, v. 94, n. 2, p. 566–572, 2007. Citado 2 vezes nas páginas 1 e 7.
- Embrapa. *Rotação de Culturas na Região Central do Brasil*. 2021. Agência Embrapa de Informação Tecnológica. Acesso em: 06 jun. 2026. Disponível em: <<https://www.embrapa.br/agencia-de-informacao-tecnologica/cultivos/soja/producao/rotacao-de-culturas/rotacao-de-culturas-na-regiao-central-do-brasil>>. Citado 5 vezes nas páginas 2, 3, 4, 19 e 37.
- EVANS, E. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston: Addison-Wesley Professional, 2003. Citado na página 16.
- FRANCHINI, J. C. et al. Importância da rotação de culturas para a produção agrícola sustentável no paraná. Londrina: Embrapa Soja, 2011., 2011. Citado 2 vezes nas páginas 1 e 2.
- GOLDBARG, M.; GOLDBARG, E. *Grafos: conceitos, algoritmos e aplicações*. [S.l.]: Elsevier, 2012. Citado 3 vezes nas páginas 4, 5 e 6.
- Google LLC. *OR-Tools: Google's Software Suite for Combinatorial Optimization*. 2023. Google Developers. Acesso em: 06 jun. 2026. Disponível em: <<https://developers.google.com/optimization>>. Citado 2 vezes nas páginas 14 e 37.
- Google LLC. *Angular Material — UI component infrastructure and Material Design components*. 2024. Angular Material Documentation. Acesso em: 06 jun. 2026. Disponível em: <<https://material.angular.io/>>. Citado na página 13.

- Google LLC. *Angular — The web development framework for building the future*. 2025. Angular Documentation. Acesso em: 06 jun. 2026. Disponível em: <<https://angular.dev/>>. Citado na página 13.
- HAZELL, P. B. R. A linear alternative to quadratic and semivariance programming for farm planning under uncertainty. *American Journal of Agricultural Economics*, Oxford University Press, v. 53, n. 1, p. 53–62, 1971. Citado na página 10.
- HILLIER, F. S. *Introduction to operations research*. [S.l.]: McGrawHill, 2005. Citado na página 6.
- JONES, M.; BRADLEY, J.; SAKIMURA, N. *JSON Web Token (JWT)*. 2015. IETF RFC 7519. Acesso em: 06 jun. 2026. Disponível em: <<https://datatracker.ietf.org/doc/html/rfc7519>>. Citado na página 16.
- LEANDRO, H. M.; ASMUS, G. L. Rotação e sucessão de culturas para o manejo do nematoide reniforme em área de produção de soja. *Ciência Rural*, Universidade Federal de Santa Maria, v. 45, n. 6, p. 945–950, 2015. Citado na página 3.
- MariaDB Foundation. *MariaDB Server Documentation*. 2023. MariaDB KnowledgeBase. Acesso em: 06 jun. 2026. Disponível em: <<https://mariadb.com/kb/en/documentation/>>. Citado na página 13.
- MARTIN, R. C. Design principles and design patterns. *Object Mentor*, v. 1, n. 34, p. 597, 2000. Citado na página 16.
- Microsoft. *Entity Framework Core Documentation*. 2024. Microsoft Learn. Acesso em: 06 jun. 2026. Disponível em: <<https://learn.microsoft.com/en-us/ef/core/>>. Citado na página 14.
- Microsoft. *.NET 10 — Official release notes and documentation*. 2024. Microsoft Learn. Acesso em: 06 jun. 2026. Disponível em: <<https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-10/overview>>. Citado na página 13.
- NETTO, P. O. B. *Grafos: teoria, modelos, algoritmos*. [S.l.]: Editora Blucher, 2003. Citado 2 vezes nas páginas 4 e 5.
- PRADO, D. *Programação linear*. [S.l.]: Falconi Editora, 2016. v. 1. Citado na página 7.
- SALASSI, M. E.; DELIBERTO, M. A.; GUIDRY, K. M. Economically optimal crop sequences using risk-adjusted network flows: Modeling cotton crop rotations in the southeastern united states. *Agricultural Systems*, Elsevier, v. 118, p. 33–40, 2013. Citado 11 vezes nas páginas , 1, 7, 8, 10, 11, 12, 30, 31, 32 e 37.
- SILVA, M. A. et al. Sistema de plantio direto e rotação de culturas no cerrado. *Research, Society and Development*, v. 11, n. 13, p. e376111335568, 2022. Disponível em: <<https://rsdjournal.org/index.php/rsd/article/view/35568>>. Citado 3 vezes nas páginas 1, 2 e 3.
- SKINNER, J. *FluentValidation Documentation*. 2024. FluentValidation. Acesso em: 06 jun. 2026. Disponível em: <<https://docs.fluentvalidation.net/>>. Citado na página 17.

SOUSA, W. *CropRotation: sistema web para planejamento de rotação de culturas*. 2026. Repositório de código-fonte. GitHub. Disponível em: <<https://github.com/willpSousa/CropRotation>>. Acesso em: 29 jun. 2026. Citado na página 28.

SZWARCFITER, J. L. *Teoria computacional de grafos: Os Algoritmos*. [S.l.]: Elsevier Brasil, 2018. Citado na página 4.

TAUER, L. W. Target MOTAD. *American Journal of Agricultural Economics*, Oxford University Press, v. 65, n. 3, p. 606–610, 1983. Citado na página 10.