

METODOLOGIA QUANTITATIVA PARA ANÁLISE DO NEXO ÁGUA-ENERGIA-ALIMENTO EM BACIAS HIDROGRÁFICAS SOB CENÁRIOS HISTÓRICOS E FUTUROS

TAMIRES LIMA DA SILVA¹; RODRIGO MÁXIMO SÁNCHEZ ROMÁN²; HESSAM S. SARJOUGHIAN³ E MOSTAFA D. FARD⁴

¹ Engenharia rural e socioeconomia, Universidade Estadual Paulista (Unesp), Faculdade de Ciências Agrônomicas, Botucatu, Av. Universitária, 3780, Altos do Paraíso, 18610-034, Botucatu, SP, Brasil, tamireslsilva@gmail.com

² Engenharia rural e socioeconomia, Universidade Estadual Paulista (Unesp), Faculdade de Ciências Agrônomicas, Botucatu, Av. Universitária, 3780, Altos do Paraíso, 18610-034, Botucatu, SP, Brasil, rodrigo.roman@unesp.br

³ Ira A. Fulton Schools of Engineering, School of Computing and Augmented Intelligence, Arizona State University (ASU), S Mill Avenue, 699, 85287-8809, Tempe, AZ, Estados Unidos da América, hessam.Sarjoughian@asu.edu

⁴ Ira A. Fulton Schools of Engineering, School of Computing and Augmented Intelligence, Arizona State University (ASU), S Mill Avenue, 699, 85287-8809, Tempe, AZ, Estados Unidos da América, sderakhs@asu.edu

1 RESUMO

O uso integrado de recursos hídricos, energéticos e alimentares é essencial para a gestão sustentável de bacias hidrográficas. Este estudo desenvolveu uma metodologia quantitativa para modelar o nexo água-energia-alimento (AEA) em bacias hidrográficas, considerando tanto condições históricas quanto projeções futuras. A metodologia proposta utiliza modelos criados nos programas “*Water Evaluation and Planning*” System (WEAP) e “*Low Emissions Analysis Platform*” (LEAP), os quais foram acoplados para troca de dados por meio do WEAP-KIB-LEAP *framework*. Para exemplificar o uso da metodologia, foi realizada uma aplicação nas bacias hidrográficas dos rios Piracicaba, Capivari e Jundiaí (PCJ). Os resultados demonstraram que a ferramenta permite a simulação do nexo AEA, oferecendo suporte valioso para o planejamento sustentável dos recursos presentes nas bacias PCJ, podendo contribuir com o alcance dos Objetivos de Desenvolvimento Sustentável (ODS). Recomenda-se a aplicação do *framework* em diferentes contextos regionais para ampliar sua validação e aplicabilidade.

Palavras-chave: WEAP, LEAP, WEAP-KIB-LEAP *framework*, nexo AEA.

SILVA, T. L.; SÁNCHEZ-ROMÁN, R. M.; SARJOUGHIAN, H. S.; FARD, M. D.
QUANTITATIVE METHODOLOGY FOR ANALYZING THE WATER-ENERGY-FOOD NEXUS IN RIVER BASINS UNDER HISTORICAL AND FUTURE SCENARIOS

2 ABSTRACT

The integrated use of water, energy, and food resources is essential for the sustainable management of river basins. This study developed a quantitative methodology to model the water–energy–food (WEF) nexus in river basins, considering both historical conditions and

future projections. The proposed methodology uses models created in the “Water Evaluation and Planning” System (WEAP) and “Low Emissions Analysis Platform” (LEAP) programs, which were coupled for data exchange through the WEAP-KIB-LEAP framework. To exemplify the use of the methodology, an application was carried out in the Piracicaba, Capivarí and Jundiá (PCJ) river basins. The results showed that the tool allows the WEF nexus to be simulated, offering valuable support for the sustainable planning of the resources present in the PCJ basins, and can contribute to achieving the Sustainable Development Goals (SDGs). It is recommended that the framework be applied in different regional contexts to expand its validation and applicability.

Keywords: WEAP, LEAP, WEAP-KIB-LEAP framework, WEF nexus.

3 INTRODUÇÃO

O planejamento eficaz dos recursos hídricos, contemplando interações entre sistemas naturais e sociais, é essencial. Nesse sentido, abordagens como a modelagem do nexos água-energia-alimento (AEA) tornam-se especialmente relevantes, pois exploram inter-relações entre esses elementos considerando tanto impactos socioeconômicos como impactos ambientais.

Diversos modelos têm sido desenvolvidos para mapear o nexos AEA em bacias hidrográficas. Um exemplo é a abordagem DAFNE (*Decision-Analytic Framework to explore the water-energy-food NEexus*), de Koundouri e Papadaki (2020), que explora as inter-relações entre os recursos água, energia e alimento em bacias hidrográficas transfronteiriças.

Outro modelo interessante é o *Water-Energy-Food Nexus Simulation Model* (WEFSiM), de Wicaksono e Kang (2019), que utilizou Dinâmica de Sistemas para equilibrar a alocação de recursos entre os setores de água, energia e alimentos, fornecendo simulações detalhadas.

Yilin *et al.* (2024) modelaram o nexos AEA na bacia hidrográfica do “Yellow River” e, por meio do método “*generalized Divisia index*”, identificaram os principais fatores que influenciaram alterações espaciais e

temporais no uso da água nos setores alimento e energia.

Os modelos citados são exemplos de modelagens quantitativas no nexos AEA. Compreender quantitativamente as inter-relações entre poluição atmosférica, clima, disponibilidade hídrica, energia e produção de alimentos pode apoiar uma gestão mais eficiente dos recursos hídricos de bacias hidrográficas. Além disso, pode apoiar políticas públicas voltadas ao alcance dos Objetivos de Desenvolvimento Sustentável (ODS) 2, 6, 7, 11, 12, 13 e 15.

Diante do exposto, o objetivo desta pesquisa foi desenvolver uma metodologia quantitativa para analisar o nexos água-energia-alimento em bacias hidrográficas, considerando condições históricas e projeções futuras.

4 MATERIAL E MÉTODOS

A modelagem do nexos AEA de forma quantitativa ocorreu por meio do uso dos programas WEAP (“*Water Evaluation and Planning*” System), que permite a criação de modelos hidrológicos holísticos, e o LEAP (“*Low Emissions Analysis Platform*”), que permite a criação de modelos que relacionam produção de energia, demanda energética e emissões atmosféricas de gases de efeito estufa. Os programas WEAP e LEAP foram

desenvolvidos e são disponibilizados pelo Instituto Ambiental de Estocolmo (Heaps, 2022; Sieber, 2023). Instituições públicas, organizações sem fins lucrativos e pesquisadores de países em desenvolvimento têm acesso gratuito aos programas.

O WEAP foi conectado ao LEAP por meio de um Modelo de Interação (MI). O MI conceitualizado por Fard e Sarjoughian (2020) utiliza a abordagem de modelagem *Knowledge Interchange Broker* (KIB), resultando no WEAP-KIB-LEAP *framework*. O algoritmo escrito em JavaScript permite o acoplamento e a troca de informações entre os programas WEAP e LEAP, que foram compartimentalizados, recebendo as denominações C-WEAP e C-LEAP. O C-WEAP transfere e recebe dados do C-LEAP

via Web API (*Application Programming Interface*) por meio do Node.JS.

O WEAP-KIB-LEAP *framework* permite a criação de interações com diferentes escalas temporais (diária, mensal ou anual). No modelo criado como exemplo para as bacias hidrográficas dos rios Piracicaba, Capivari e Jundiá (PCJ), a escala temporal dos dados no WEAP foi mensal, enquanto os dados de produção, consumo de energia e emissão de gases de efeito estufa no LEAP apresentaram escala anual.

No estabelecimento de uma conexão entre o C-WEAP e o C-LEAP para a modelagem do nexa AEA, os seguintes passos presentes no Quadro 1 devem ser seguidos.

Quadro 1. Passos para o estabelecimento de uma conexão entre o C-WEAP e C-LEAP por meio do WEAP-KIB-LEAP *framework*

1.	definição das portas dos módulos (Ex.: InputFromWEAP, OutputToLEAP);
2.	adição das portas dos módulos (addInput, addOutput);
3.	definição das portas de transformação dos dados (Ex.: WEAP_LEAP_Transformation);
4.	adição da transformação (addTransformation);
5.	definição do acoplamento entre as portas dos módulos e da transformação (Coupling c1 e Coupling c2);
6.	adição do acoplamento (addCoupling).

A Figura 1 apresenta a estrutura geral de uma conexão entre o C-WEAP e C-LEAP por meio do WEAP-KIB-LEAP *framework*.

Cada parte do algoritmo está relacionada aos passos (1 ao 6) a serem seguidos.

Figura 1. Estrutura geral para uma conexão entre o C-WEAP e C-LEAP por meio do WEAP-KIB-LEAP *framework*

```

                                Passo 1
InputFromWEAP in1 = new InputFromWEAP("in1", new
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.DemandSite, "Componente Name" ,
PortTypes.Output, "Variable Name", "Scenario Name"));

OutputToLEAP out1 = new OutputToLEAP("out1", new
LeapParameters(LEAP_PROJECT_NAME,
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Component Name",
kib.few.leap.utilities.enums.PortTypes.Input, "Variable Name", "Scenario Name"));

                                Passo 2

addInput(in1);
addOutput(out1);

                                Passo 3

WEAP_LEAP_Transformation tran1 = new WEAP_LEAP_Transformation("W-L-Tran1");

                                Passo 4

addTransformation(tran1);

                                Passo 5

Coupling c1 = new Coupling(in1, tran1.getInput("in1"));
Coupling c2 = new Coupling(tran1.getOutput("out1"), out1);

                                Passo 6

addCoupling(c1);
addCoupling(c2);

```

A denominação WEAP_PROJECT_NAME e LEAP_PROJECT_NAME (Figura 1) refere-se ao nome do modelo criado no WEAP e no LEAP, respectivamente, do qual pretende-se obter o dado em questão. Para as condições históricas no exemplo de aplicação da metodologia nas bacias PCJ, os projetos no WEAP e LEAP receberam a mesma

denominação, sendo nomeados como “Cenário BAU”. Já para as projeções futuras, o projeto LEAP foi denominado “Cenário RCP” e o WEAP “Cenário RCP4.5 e RCP8.5”.

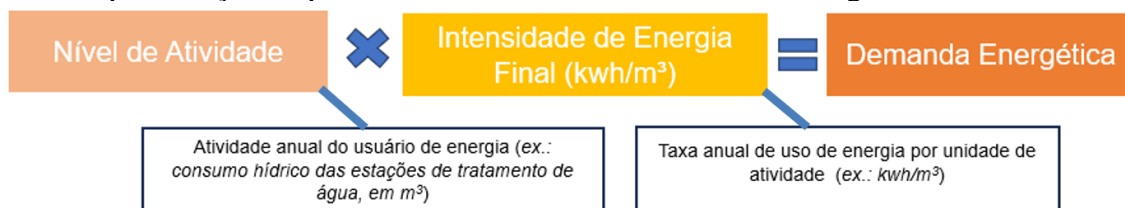
Os *ComponentTypes* (Figura 1) referem-se ao tipo de entidade/componente do sistema. As possibilidades para o WEAP são apresentadas no Quadro 2.

Quadro 2. Tipos de entidade/componente (*ComponentTypes*) do WEAP

Nó	<i>DemandSite</i> (Demanda), <i>Groundwater</i> (Água subterrânea), <i>Reservoir</i> (Reservatório), <i>OtherSupplies</i> (Outras fontes hídricas), <i>WastewaterTreatment</i> (Estação de Tratamento de Esgoto) e <i>Catchment</i> (Sub-bacia hidrográfica);
Link	<i>Transmissions</i> (Link de transmissão), <i>Runoffs</i> (Escoamento) e <i>ReturnFlows</i> (Vazão de retorno)
Flow	<i>Rivers</i> (Rio) e <i>Diversions</i> (Desvios)

No LEAP, os *ComponentTypes* podem ser do tipo *Resource* (Recurso), *Transformations* (Transformações), *Process* (Processo), *Demand* (Demanda) ou *Effect* (Efeito). O recurso refere-se à fonte utilizada na geração de energia (hídrica, solar, eólica, biomassa, etc.). A transformação tem ligação com tipo de energia a ser gerada, no caso das bacias PCJ o tipo de energia modelado foi energia elétrica, e os processos envolvidos (características das usinas de geração de energia). A demanda diz respeito ao consumo de energia e o efeito é a emissão de poluentes.

Antes de prosseguir, é importante entender o procedimento de cálculo de demanda energética pelo programa LEAP a fim de estabelecer relações entre as variáveis. O cálculo envolve a multiplicação dos fatores: “nível de atividade da demanda” e “intensidade de energia final” (Figura 2). Exemplificando, supondo que no ano de 2014, o nível de atividade das Estações de Tratamento de Água presentes na bacia hidrográfica tenha sido de 1.000.000 m³ com intensidade de energia final igual a 0,3 kwh/m³, a demanda energética dessas Estações de Tratamento de Água no ano de 2014 seria de 300.000 kwh.

Figura 2. Representação do procedimento de cálculo da demanda energética no LEAP

Em *ComponentName* (Figura 1), define-se o nome do componente. Enquanto em *PortTypes* define-se o tipo de porta, a mesma pode ser um dado de entrada (*Input*) ou saída (*Output*) do modelo criado no WEAP ou no LEAP. Já em *VariableName*, define-se o nome das variáveis com as quais deseja-se efetuar a conexão entre os programas WEAP e LEAP.

Por fim, em *ScenarioName* (Figura 1), define-se o nome do cenário. O WEAP, assim

como o LEAP, permite a criação de cenários baseados em suposições, apesar disso, ambos possuem cenários predefinidos que não apresentam possibilidade de exclusão. No WEAP, eles recebem a denominação de *Current Accounts*, que é referente ao primeiro ano da série histórica, e *Reference*, que engloba os demais anos da série. Ao passo que no LEAP, os cenários predefinidos são *Current Accounts* e *Baseline*.

Para a execução do código do MI, além da instalação dos programas WEAP e LEAP, também é necessária a instalação dos programas: NodeJS, Visual Studio Code, Java 8, Java (última versão lançada), Python 2.7 e Microsoft visual C++. Adicionalmente, é imprescindível a instalação das seguintes extensões no Visual Studio Code: C/C++, Intellicode API, Extension Pack for Java; JavaScript (ES6) code Snippets; Node.js Exec; Node.js code Snippets; Prettier-code formatter; TSLint Snippets; TSLint Vue; TSLint Vue-TSX; e TypeScript + Webpack Problem.

As interações estabelecidas por meio do MI nas bacias PCJ (Figura 3) foram:

- O volume de água demandado nas bacias PCJ para o abastecimento urbano e rural, em metros cúbicos (m³), calculado no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda das Estações de Tratamento de Água (ETAs).;
- O volume de esgoto tratado nas Estações de Tratamento de Esgoto (ETEs), em m³, dado de entrada no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda ETEs;
- O volume de água demandado pela irrigação por superfície, em m³, calculado no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda irrigação por superfície;
- O volume de água demandado pela irrigação por aspersão convencional, em m³, calculado no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda aspersão convencional;
- O volume de água demandado pela irrigação por sistema autopropelido, em m³, calculado no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda autopropelido;
- O volume de água demandado pela irrigação por gotejamento, em m³, calculado no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda gotejamento;
- A área irrigada por microaspersão, em hectares, dado de entrada no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda microaspersão;
- A produção de cana-de-açúcar na bacia hidrográfica, em toneladas, dado de entrada no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda colheitadeiras;
- A área plantada com soja na bacia hidrográfica, em hectares, dado de entrada no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente às demandas: soja gradeamento, soja escarificador, soja semeadora, soja pulverização e soja colheita. Neste caso, avaliou-se o consumo de energia (diesel) para o cultivo da soja, considerando-se diferentes técnicas de preparo do solo antecedentes ao plantio, as quais foram:
 - a) Cultivo mínimo, técnica na qual antes do plantio há uso do escarificador (soja escarificador), seguido pelo uso da semeadora (soja semeadora);
 - b) Plantio direto, técnica na qual antes do plantio há somente uso da semeadora (soja semeadora); e
 - c) Cultivo convencional, técnica na qual antes do plantio há uso de grade (soja gradeamento), seguido pelo uso da semeadora (soja semeadora).
 Além disso, foram somadas as demandas energéticas durante a prática de

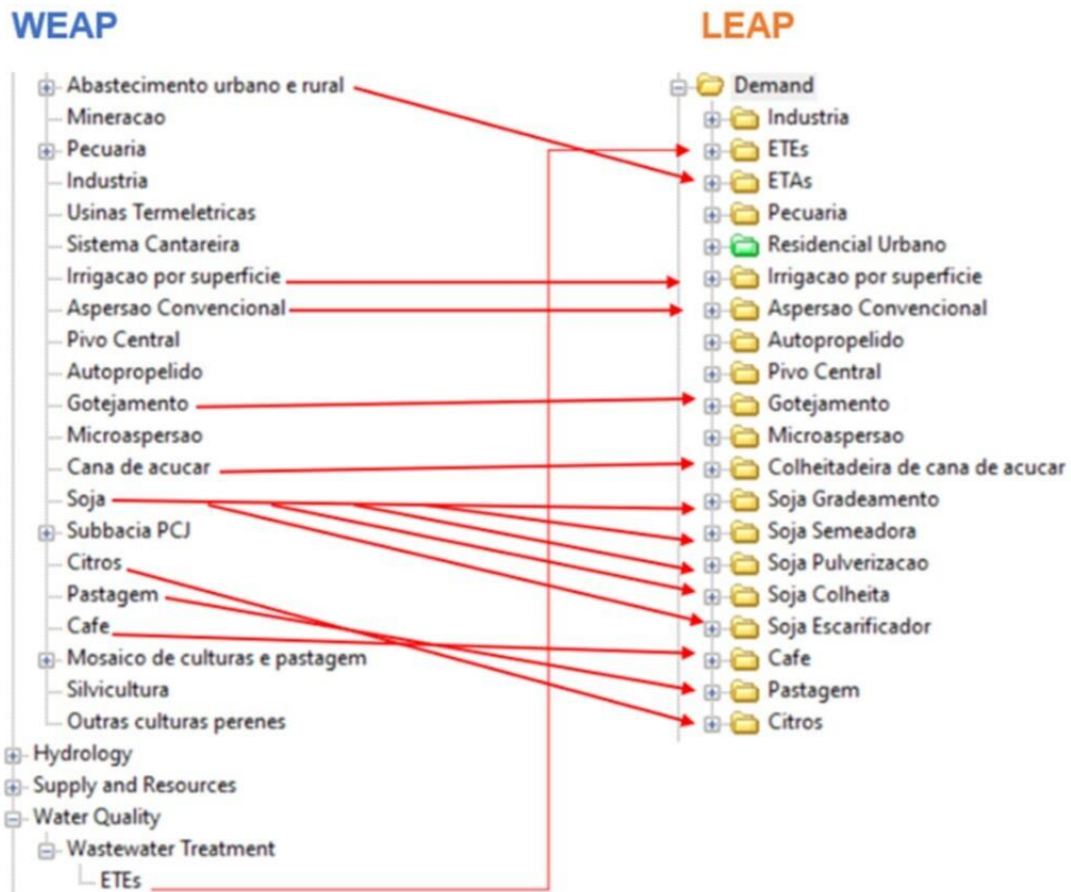
pulverização (soja pulverização) e na colheita (soja colheita).

- A área plantada com café na bacia hidrográfica, em hectares, dado de entrada no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda café. Neste caso, avaliou-se o consumo de energia (diesel) por maquinários considerando todo o ciclo da cultura.
- A área plantada com citros na bacia hidrográfica, em hectares, dado de entrada no WEAP, foi utilizado como

dado de entrada para o nível de atividade no LEAP referente à demanda citros, considerando-se todo o consumo de energia (diesel) dos maquinários utilizados durante o manejo da cultura.

- A área plantada com pastagem na bacia hidrográfica, em hectares, dado de entrada no WEAP, foi utilizado como dado de entrada para o nível de atividade no LEAP referente à demanda pastagem, abrangendo-se o consumo de energia (diesel) dos maquinários empregados durante a condução da pastagem.

Figura 3. Interações estabelecidas entre o WEAP e o LEAP



Fonte: Silva (2023)

É importante destacar que tanto a água tratada nas ETAs como a água residuária proveniente das ETEs, são fontes

hídricas com potencial de ser empregadas na irrigação.

5 RESULTADOS E DISCUSSÃO

O MI para a modelagem quantitativa do nexo AEA, utilizando os programas WEAP, LEAP e o WEAP-KIB-LEAP *framework*, é apresentado a seguir. Destaca-

se que nos projetos WEAP e LEAP, manteve-se as denominações para os elementos inseridos, tanto nas condições históricas quanto nas projetadas. As projeções de demanda e as mudanças no uso e cobertura da terra foram realizadas com base em dados históricos.

```
package models.NexoAEA.modules;
import models.NexoAEA.transformations.LEAP_WEAP_Transformation;
import models.NexoAEA.transformations.WEAP_LEAP_Transformation;
import kib.core.domain.Coupling;
import kib.core.domain.Module;
import kib.few.leap.ports.InputFromLEAP;
import kib.few.leap.ports.OutputToLEAP;
import kib.few.leap.ports.params.LeanParameters;
import kib.few.weap.ports.InputFromWEAP;
import kib.few.weap.ports.OutputToWEAP;
import kib.few.weap.ports.params.WeapParameters;
import kib.few.weap.utilities.enums.ComponentTypes;
import kib.few.weap.utilities.enums.PortTypes;
import kib.utilities.exceptions.kib.KIBModelException;
public class SimpleModule extends Module {
private final static String WEAP_PROJECT_NAME = "Cenario BAU";
private final static String LEAP_PROJECT_NAME = "Cenario BAU";
public SimpleModule(String name) throws KIBModelException {
super(name);
initialization();
}
private void initialization() throws KIBModelException {
InputFromWEAP in1 = new InputFromWEAP("in1", new
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.DemandSite,
"Abastecimento urbano e rural" , PortTypes.Output, "Water Demand", "Reference"));

OutputToLEAP out1 = new OutputToLEAP("out1", new
LeapParameters(LEAP_PROJECT_NAME,
kib.few.leap.utilities.enums.ComponentTypes.Demand, "ETAs",
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));

InputFromWEAP in2 = new InputFromWEAP("in2", new
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.DemandSite,
"Irigacao por superficie" , PortTypes.Output, "Water Demand", "Reference"));

OutputToLEAP out2 = new OutputToLEAP("out2", new
LeapParameters(LEAP_PROJECT_NAME,
```



```
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Irigacao por superficie",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in3 = new InputFromWEAP("in3", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.DemandSite,  
"Aspersao Convencional", PortTypes.Output, "Water Demand", "Reference"));
```

```
OutputToLEAP out3 = new OutputToLEAP("out3", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Aspersao Convencional",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in4 = new InputFromWEAP("in4", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.DemandSite,  
"Autopropelido", PortTypes.Output, "Water Demand", "Reference"));  
OutputToLEAP out4 = new OutputToLEAP("out4", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Autopropelido",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in5 = new InputFromWEAP("in5", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.DemandSite, "Pivo  
Central", PortTypes.Output, "Water Demand", "Reference"));
```

```
OutputToLEAP out5 = new OutputToLEAP("out5", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Pivo Central",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in6 = new InputFromWEAP("in6", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.DemandSite,  
"Gotejamento", PortTypes.Output, "Water Demand", "Reference"));
```

```
OutputToLEAP out6 = new OutputToLEAP("out6", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Gotejamento",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in7 = new InputFromWEAP("in7", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.DemandSite,  
"Microaspersao", PortTypes.Input, "Annual Activity Level", "Reference"));
```

```
OutputToLEAP out7 = new OutputToLEAP("out7", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Microaspersao",
```

```
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in8 = new InputFromWEAP("in8", new  
WeapParameters(WEAP_PROJECT_NAME,  
ComponentTypes.WastewaterTreatment, "ETEs" , PortTypes.Input, "Daily Capacity",  
"Reference"));
```

```
OutputToLEAP out8 = new OutputToLEAP("out8", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "ETEs",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));  
InputFromWEAP in9 = new InputFromWEAP("in9", new  
WeapParameters(WEAP_PROJECT_NAME,  
ComponentTypes.WastewaterTreatment, "ETEs" , PortTypes.Input, "Daily Capacity",  
"Current Accounts"));  
OutputToLEAP out9= new OutputToLEAP("out9", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "ETEs",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Current Accounts"));
```

```
InputFromWEAP in10 = new InputFromWEAP("in10", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment, "Cana  
de acucar" , PortTypes.Input, "Production", "Reference"));
```

```
OutputToLEAP out10= new OutputToLEAP("out10", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Colheitadeira de cana de  
acucar", kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in11 = new InputFromWEAP("in11", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment, "Cana  
de acucar" , PortTypes.Input, "Production", "Current Accounts"));
```

```
OutputToLEAP out11= new OutputToLEAP("out11", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Colheitadeira de cana de  
acucar", kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Current  
Accounts"));
```

```
InputFromWEAP in12 = new InputFromWEAP("in12", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment, "Cafe" ,  
PortTypes.Input, "Area", "Reference"));
```

```
OutputToLEAP out12= new OutputToLEAP("out12", new  
LeapParameters(LEAP_PROJECT_NAME,
```

```
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Cafe",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in13 = new InputFromWEAP("in13", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment, "Cafe" ,  
PortTypes.Input, "Area", "Current Accounts"));
```

```
OutputToLEAP out13= new OutputToLEAP("out13", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Cafe",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Current Accounts"));
```

```
InputFromWEAP in14 = new InputFromWEAP("in14", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment,  
"Pastagem" , PortTypes.Input, "Area", "Reference"));  
OutputToLEAP out14= new OutputToLEAP("out14", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Pastagem",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in15 = new InputFromWEAP("in15", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment,  
"Pastagem" , PortTypes.Input, "Area", "Current Accounts"));
```

```
OutputToLEAP out15= new OutputToLEAP("out15", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Pastagem",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Current Accounts"));
```

```
InputFromWEAP in16 = new InputFromWEAP("in16", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment, "Citros" ,  
PortTypes.Input, "Area", "Reference"));
```

```
OutputToLEAP out16= new OutputToLEAP("out16", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Citros",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
InputFromWEAP in17 = new InputFromWEAP("in17", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment, "Citros" ,  
PortTypes.Input, "Area", "Current Accounts"));
```

```
OutputToLEAP out17= new OutputToLEAP("out17", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Citros",
```

```
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Current Accounts"));
```

```
InputFromWEAP in18 = new InputFromWEAP("in18", new  
WeapParameters(WEAP_PROJECT_NAME, ComponentTypes.Catchment, "Soja" ,  
PortTypes.Input, "Area", "Reference"));
```

```
OutputToLEAP out18= new OutputToLEAP("out18", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Soja Gradeamento",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
OutputToLEAP out19= new OutputToLEAP("out19", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Soja Semeadora",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));  
OutputToLEAP out20= new OutputToLEAP("out20", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Soja Pulverizacao",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
OutputToLEAP out21= new OutputToLEAP("out21", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Soja Colheita",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));
```

```
OutputToLEAP out22= new OutputToLEAP("out22", new  
LeapParameters(LEAP_PROJECT_NAME,  
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Soja Escarificador",  
kib.few.leap.utilities.enums.PortTypes.Input, "Activity Level", "Baseline"));  
in1.setPriority(1);
```

```
addInput(in1);  
addInput(in2);  
addInput(in3);  
addInput(in4);  
addInput(in5);  
addInput(in6);  
addInput(in7);  
addInput(in8);  
addInput(in9);  
addInput(in10);  
addInput(in11);  
addInput(in12);  
addInput(in13);  
addInput(in14);
```

```
addInput(in15);
addInput(in16);
addInput(in17);
addInput(in18);
```

```
addOutput(out1);
addOutput(out2);
addOutput(out3);
addOutput(out4);
addOutput(out5);
addOutput(out6);
addOutput(out7);
addOutput(out8);
addOutput(out9);
addOutput(out10);
addOutput(out11);
addOutput(out12);
addOutput(out13);
addOutput(out14);
addOutput(out15);
addOutput(out16);
addOutput(out17);
addOutput(out18);
addOutput(out19);
addOutput(out20);
addOutput(out21);
addOutput(out22);
```

```
WEAP_LEAP_Transformation tran1 = new WEAP_LEAP_Transformation("W-LTran1");
addTransformation(tran1);
Coupling c1 = new Coupling(in1, tran1.getInput("in1"));
Coupling c2 = new Coupling(tran1.getOutput("out1"), out1);
```

```
WEAP_LEAP_Transformation tran2 = new WEAP_LEAP_Transformation("W-LTran2");
addTransformation(tran2);
Coupling c3 = new Coupling(in2, tran2.getInput("in1"));
Coupling c4 = new Coupling(tran2.getOutput("out1"), out2);
```

```
WEAP_LEAP_Transformation tran3 = new WEAP_LEAP_Transformation("W-LTran3");
addTransformation(tran3);
Coupling c5 = new Coupling(in3, tran3.getInput("in1"));
Coupling c6 = new Coupling(tran3.getOutput("out1"), out3);
```

```
WEAP_LEAP_Transformation tran4 = new WEAP_LEAP_Transformation("W-LTran4");
addTransformation(tran4);
```

```
Coupling c7 = new Coupling(in4, tran4.getInput("in1"));
Coupling c8 = new Coupling(tran4.getOutput("out1"), out4);

WEAP_LEAP_Transformation tran5 = new WEAP_LEAP_Transformation("W-LTran5");
addTransformation(tran5);
Coupling c9 = new Coupling(in5, tran5.getInput("in1"));
Coupling c10 = new Coupling(tran5.getOutput("out1"), out5);
WEAP_LEAP_Transformation tran6 = new WEAP_LEAP_Transformation("W-LTran6");
addTransformation(tran6);
Coupling c11 = new Coupling(in6, tran6.getInput("in1"));
Coupling c12 = new Coupling(tran6.getOutput("out1"), out6);

WEAP_LEAP_Transformation tran7 = new WEAP_LEAP_Transformation("W-LTran7");
addTransformation(tran7);
Coupling c13 = new Coupling(in7, tran7.getInput("in1"));
Coupling c14 = new Coupling(tran7.getOutput("out1"), out7);

WEAP_LEAP_Transformation tran8 = new WEAP_LEAP_Transformation("W-LTran8");
addTransformation(tran8);
Coupling c15 = new Coupling(in8, tran8.getInput("in1"));
Coupling c16 = new Coupling(tran8.getOutput("out1"), out8);

WEAP_LEAP_Transformation tran9 = new WEAP_LEAP_Transformation("W-LTran9");
addTransformation(tran9);
Coupling c17 = new Coupling(in9, tran9.getInput("in1"));
Coupling c18 = new Coupling(tran9.getOutput("out1"), out9);

WEAP_LEAP_Transformation tran10 = new WEAP_LEAP_Transformation("W-LTran10");
addTransformation(tran10);
Coupling c19 = new Coupling(in10, tran10.getInput("in1"));
Coupling c20 = new Coupling(tran10.getOutput("out1"), out10);

WEAP_LEAP_Transformation tran11 = new WEAP_LEAP_Transformation("W-LTran11");
addTransformation(tran11);
Coupling c21 = new Coupling(in11, tran11.getInput("in1"));
Coupling c22 = new Coupling(tran11.getOutput("out1"), out11);

WEAP_LEAP_Transformation tran12 = new WEAP_LEAP_Transformation("W-LTran12");
addTransformation(tran12);
Coupling c23 = new Coupling(in12, tran12.getInput("in1"));
Coupling c24 = new Coupling(tran12.getOutput("out1"), out12);

WEAP_LEAP_Transformation tran13 = new WEAP_LEAP_Transformation("W-LTran13");
addTransformation(tran13);
Coupling c25 = new Coupling(in13, tran13.getInput("in1"));
```

```
Coupling c26 = new Coupling(tran13.getOutput("out1"), out13);

WEAP_LEAP_Transformation tran14 = new WEAP_LEAP_Transformation("W-LTran14");
addTransformation(tran14);
Coupling c27 = new Coupling(in14, tran14.getInput("in1"));
Coupling c28 = new Coupling(tran14.getOutput("out1"), out14);

WEAP_LEAP_Transformation tran15 = new WEAP_LEAP_Transformation("W-LTran15");
addTransformation(tran15);
Coupling c29 = new Coupling(in15, tran15.getInput("in1"));
Coupling c30 = new Coupling(tran15.getOutput("out1"), out15);

WEAP_LEAP_Transformation tran16 = new WEAP_LEAP_Transformation("W-LTran16");
addTransformation(tran16);
Coupling c31 = new Coupling(in16, tran16.getInput("in1"));
Coupling c32 = new Coupling(tran16.getOutput("out1"), out16);

WEAP_LEAP_Transformation tran17 = new WEAP_LEAP_Transformation("W-LTran17");
addTransformation(tran17);
Coupling c33 = new Coupling(in17, tran17.getInput("in1"));
Coupling c34 = new Coupling(tran17.getOutput("out1"), out17);

WEAP_LEAP_Transformation tran18 = new WEAP_LEAP_Transformation("W-LTran18");
addTransformation(tran18);
Coupling c35 = new Coupling(in17, tran18.getInput("in1"));
Coupling c36 = new Coupling(tran18.getOutput("out1"), out18);

WEAP_LEAP_Transformation tran19 = new WEAP_LEAP_Transformation("W-LTran19");
addTransformation(tran19);
Coupling c37 = new Coupling(in17, tran19.getInput("in1"));
Coupling c38 = new Coupling(tran19.getOutput("out1"), out19);

WEAP_LEAP_Transformation tran20 = new WEAP_LEAP_Transformation("W-LTran20");
addTransformation(tran20);
Coupling c39 = new Coupling(in17, tran20.getInput("in1"));
Coupling c40 = new Coupling(tran20.getOutput("out1"), out20);

WEAP_LEAP_Transformation tran21 = new WEAP_LEAP_Transformation("W-LTran21");
addTransformation(tran21);
Coupling c41 = new Coupling(in17, tran21.getInput("in1"));
Coupling c42 = new Coupling(tran21.getOutput("out1"), out21);

WEAP_LEAP_Transformation tran22 = new WEAP_LEAP_Transformation("W-LTran22");
addTransformation(tran22);
Coupling c43 = new Coupling(in17, tran22.getInput("in1"));
```

```
Coupling c44 = new Coupling(tran22.getOutput("out1"), out22);
```

```
addCoupling(c1);  
addCoupling(c2);  
addCoupling(c3);  
addCoupling(c4);  
addCoupling(c5);  
addCoupling(c6);  
addCoupling(c7);  
addCoupling(c8);  
addCoupling(c9);  
addCoupling(c10);  
addCoupling(c11);  
addCoupling(c12);  
addCoupling(c13);  
addCoupling(c14);  
addCoupling(c15);  
addCoupling(c16);  
addCoupling(c17);  
addCoupling(c18);  
addCoupling(c19);  
addCoupling(c20);  
addCoupling(c21);  
addCoupling(c22);  
addCoupling(c23);  
addCoupling(c24);  
addCoupling(c25);  
addCoupling(c26);  
addCoupling(c27);  
addCoupling(c28);  
addCoupling(c29);  
addCoupling(c30);  
addCoupling(c31);  
addCoupling(c32);  
addCoupling(c33);  
addCoupling(c34);  
addCoupling(c35);  
addCoupling(c36);  
addCoupling(c37);  
addCoupling(c38);  
addCoupling(c39);  
addCoupling(c40);  
addCoupling(c41);  
addCoupling(c42);  
addCoupling(c43);
```



```

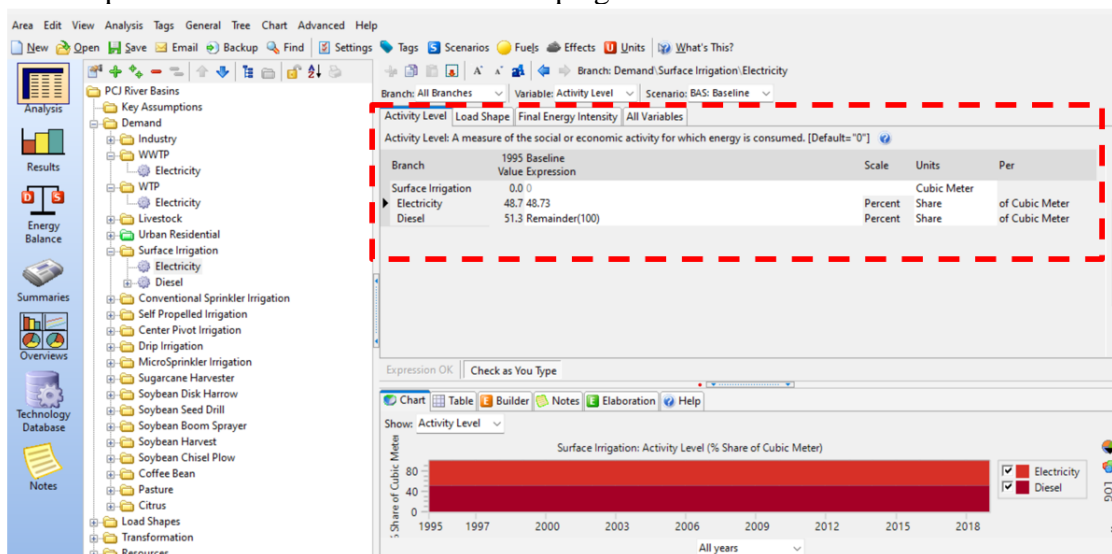
addCoupling(c44);
}
}

```

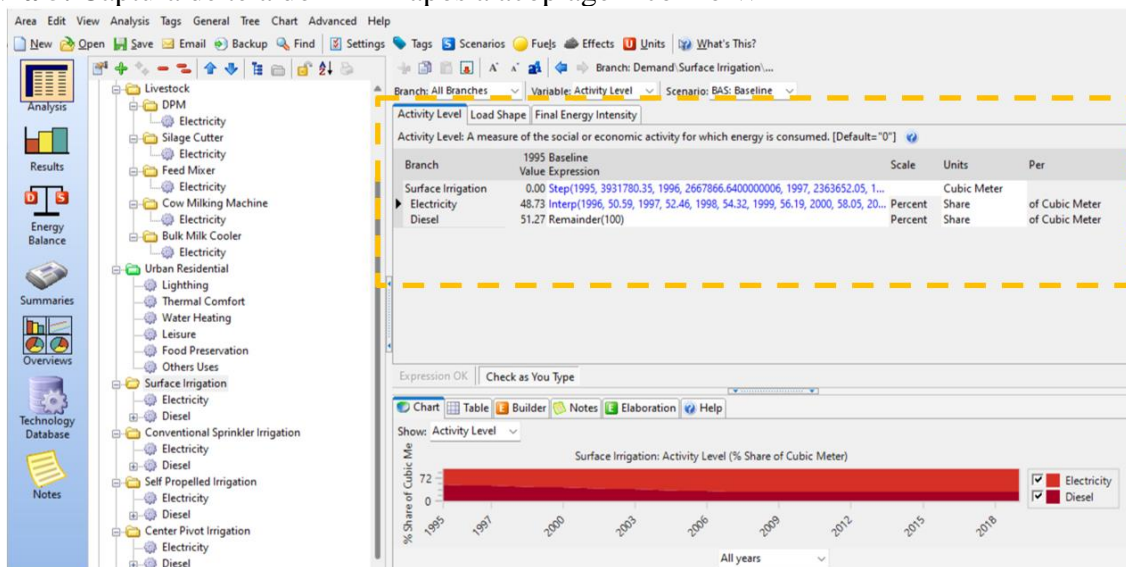
As Figuras 4 e 5 apresentam capturas de tela que mostram o estado do modelo LEAP antes e depois da sua acoplagem ao WEAP via o MI, na versão em que as variáveis estão traduzidas para o inglês. Antes desse processo, não havia dados para o

nível de atividade (*Activity Level*) da demanda de energia para a irrigação por superfície (*Surface Irrigation*) no modelo LEAP. Após a execução do MI, o campo mencionado foi preenchido com dados transferidos do WEAP para o LEAP.

Figura 4. Captura de tela do LEAP antes da acoplagem com o WEAP



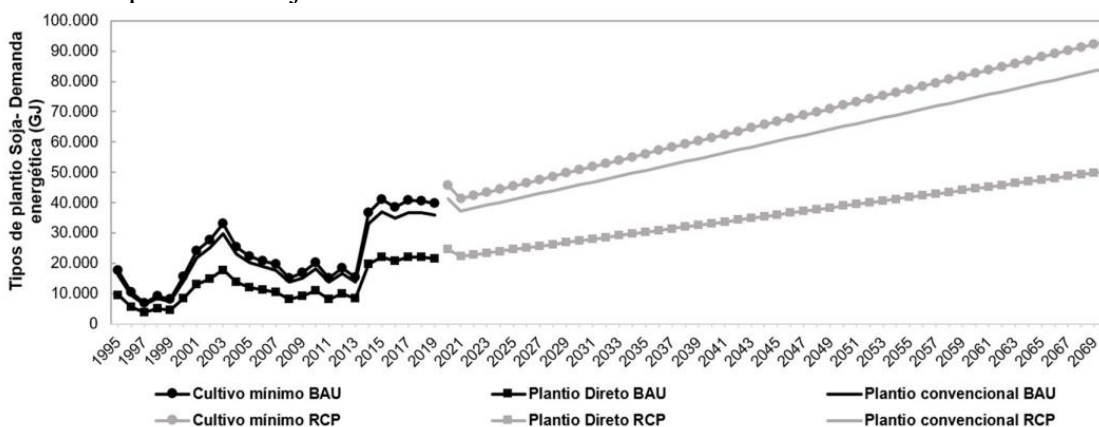
Fonte: Silva (2023)

Figura 5. Captura de tela do LEAP após a acoplagem com o WEAP

Fonte: Silva (2023)

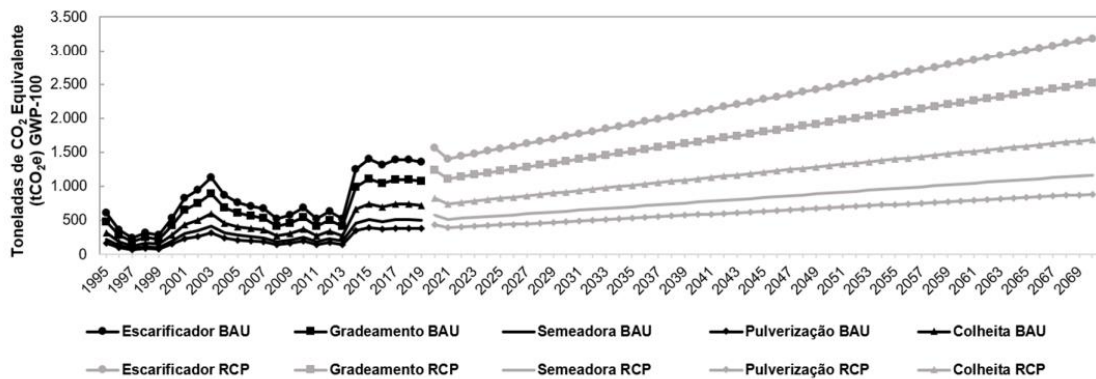
Algumas quantificações do nexso AEA nas bacias PCJ sob condições históricas e

projeções futuras podem ser visualizadas nas Figuras 6, 7, 8, 9 e 10.

Figura 6. Demanda energética, em gigajoules (GJ), referente ao uso de diesel em cada tipo de plantio de soja nos cenários BAU e RCP

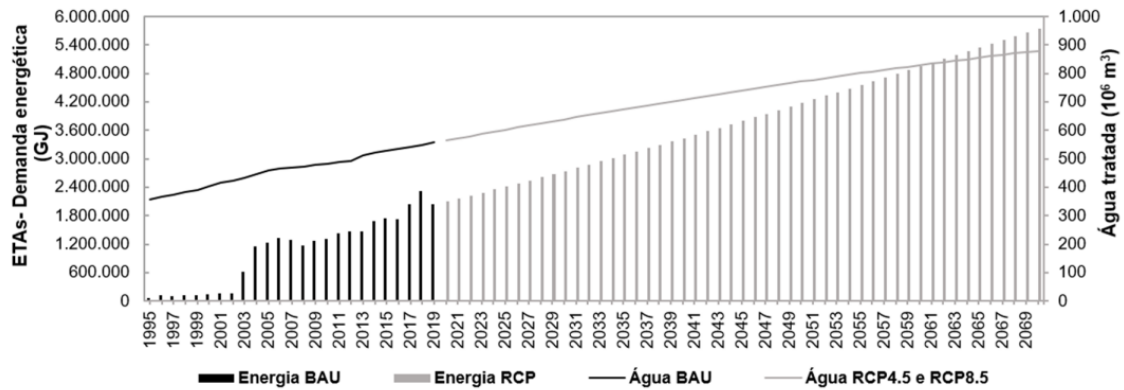
Fonte: Silva (2023)

Figura 7. Toneladas de CO₂ equivalente (GWP-100) resultante do tipo de plantio da soja nos cenários BAU e RCP



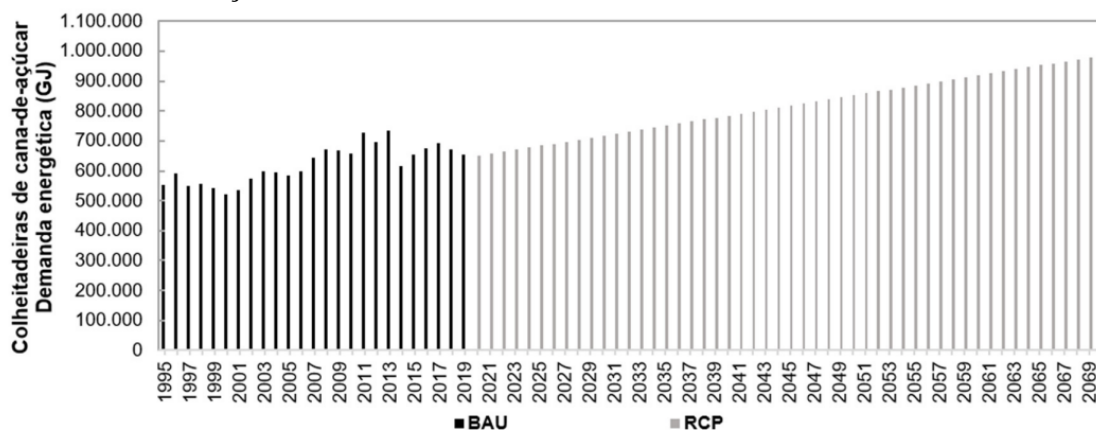
Fonte: Silva (2023)

Figura 8. Volume de água tratada e demanda energética das Estações de Tratamento de Água (ETAs) localizadas nas bacias PCJ nos cenários BAU e RCP (modelo LEAP) e BAU e RCP4.5 e RCP8.5 (modelo WEAP)



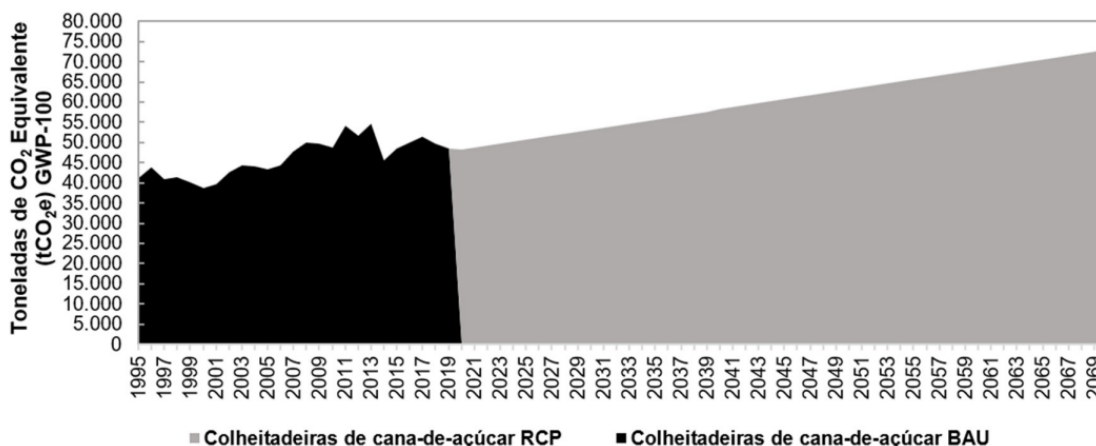
Fonte: Silva (2023)

Figura 9. Demanda energética, em gigajoules (GJ), referente ao uso de diesel em colheitadeiras de cana-de-açúcar nos cenários BAU e RCP



Fonte: Silva (2023)

Figura 10. Toneladas de CO₂ equivalente (GWP-100) resultante do uso de diesel em colheitadeiras de cana-de-açúcar nos cenários BAU e RCP



Fonte: Silva (2023)

6 CONCLUSÃO

O WEAP-KIB-LEAP *framework* mostrou-se uma ferramenta eficaz para modelar quantitativamente o nexos água-energia-alimento em bacias hidrográficas, permitindo a integração de modelos desenvolvidos no WEAP e no LEAP com diferentes escalas temporais.

A aplicação da metodologia nas bacias dos rios Piracicaba, Capivari e Jundiá, com dados históricos e projetados,

proporcionou uma visão ampla e detalhada das interações entre recursos hídricos, energéticos e alimentares. Essa abordagem oferece uma base sólida para o planejamento e a otimização do uso de recursos hídricos e energéticos, alinhada aos Objetivos de Desenvolvimento Sustentável, e representa uma contribuição relevante para a gestão sustentável de bacias hidrográficas.

Recomenda-se a aplicação do WEAP-KIB-LEAP *framework* em diferentes contextos e regiões para avaliar sua

viabilidade e potencial de adaptação a outras realidades.

7 AGRADECIMENTOS

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), pela bolsa de estudos concedida por meio do processo 143899/2019-8, que possibilitou a dedicação integral à pesquisa. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) - Código de Financiamento 001.

8 REFERÊNCIAS

KOUNDOURI, P.; PAPADAKI, L. Integrating Water-Food-Energy Nexus with Climate Services: Modelling and Assessment for a Case Study in Africa. In: KULSHRESHTHA, S. N. (ed.). **Sustainability Concept in Developing Countries**. London, UK: Intechopen, 2020. cap. 3, p. 96-116.

SILVA, T. L. **Modelagem dos impactos das mudanças climáticas nas bacias hidrográficas dos rios Piracicaba, Capivari e Jundiá considerando o nexo água-energia-alimento**. 2023. Tese (Doutorado em Irrigação e Drenagem) – Faculdade de Ciências Agrônômicas, Universidade Estadual Paulista, Botucatu, 2023.

WICAKSONO, A.; KANG, D. Nationwide simulation of water, energy, and food nexus: case study in South Korea and Indonesia. **Journal of Hydro-Environment Research**, Amsterdam, v. 22, n. 1, p. 70-87, jan. 2019. DOI: 10.1016/j.jher.2018.10.003. Disponível em: <http://dx.doi.org/10.1016/j.jher.2018.10.003>. Acesso em: 1 set. 2020.

YILIN, S.; YING, G.; YUANYUAN, G.; LANZHEN, W.; YANJUN, S. Evaluating water resources sustainability of water-scarcity basin from a scope of WEF-Nexus decomposition: the case of Yellow River Basin. **Environment, Development and Sustainability**, Amsterdam, v. 26, n. 1, p. 29583–29603, março 2024. DOI: 10.1007/s10668-024-04586-6. Disponível em: <https://doi.org/10.1007/s10668-024-04586-6>. Acesso em 1 set. 2024.