ISSN 1808-8546 (ONLINE) 1808-3765 (CD-ROM)

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

TAMIRES LIMA DA SILVA[1]; RODRIGO MÁXIMO SÁNCHEZ ROMÁN[2]; HESSAM S. SARJOUGHIAN[3] E MOSTAFA D. FARD[4]

[1] Engenharia rural e socioeconomia, Universidade Estadual Paulista (Unesp), Faculdade de Ciências Agronômicas, 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 Agronômicas, 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

Recebido em 20/09/2022 e aprovado para publicação em 09/10/2023
DOI: http://dx.doi.org/10.15809/irriga.2023v28n4p633-653

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 Jundiaí (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 INTRODUCTION

Effective water resource planning, which considers interactions between natural and social systems, is essential. In this sense, approaches such as water–energy–food (WEF) nexus modeling have become especially relevant, as they explore interrelationships between these elements while considering both socioeconomic impacts and environmental impacts.

Several models have been developed to map the ALE nexus in river basins. One example is the DAFNE approach (decision-analytic framework for exploring water, energy, and food NExus) by Koundouri and Papadaki (2020), which explores the interrelationships among water, energy, and food resources in transboundary river basins.

Another interesting model is the water-energy-food nexus simulation model (WEFSiM) by Wicaksono and Kang (2019), which uses system dynamics to balance the allocation of resources between the water, energy and food sectors, providing detailed simulations.

Yilin *et al. (2024) modeled the AEA nexus in the Yellow River* Basin and, via the *generalized Divisia index*, identified the main factors influencing spatial and temporal changes in water use in the food and energy sectors.

The models cited are examples of quantitative modeling in the ALE nexus. A quantitative understanding of the interrelationships among air pollution, climate, water availability, energy, and food production can support more efficient management of water resources in river basins. In addition, it can support public policies aimed at achieving Sustainable Development Goals (SDGs) 2, 6, 7, 11, 12, 13, and 15.

Given the above, the objective of this research was to develop a quantitative methodology to analyze the water–energy–food nexus in river basins, considering historical conditions and future projections.

4 MATERIALS AND METHODS

The quantitative modeling of the AEA nexus occurred through the use of WEAP programs *("Water Evaluation and Planning" System)*, which allows the creation of holistic hydrological models, and LEAP (*"Low Emissions Analysis Platform"*), which allows the creation of models that relate energy production, energy demand and atmospheric greenhouse gas emissions. The WEAP and LEAP programs were developed and made available by the Stockholm Environment Institute (Heaps, 2022; Sieber, 2023). Public institutions, nonprofit organizations and researchers from developing countries have free access to these programs.

WEAP was connected to LEAP through an interaction model (IM). The IM conceptualized by Fard and Sarjoughian (2020) uses the *knowledge* modeling approach *interchange broker* (KIB), resulting in the WEAP-KIB-LEAP *framework*. The algorithm written in JavaScript allows the coupling and exchange of information between the WEAP and LEAP programs, which are compartmentalized and receive the names C-WEAP and C-LEAP. C-WEAP transfers and receives data from C-LEAP via the Web API (*application programming interface*) through Node.JS.

The WEAP-KIB-LEAP *framework* allows the creation of interactions with different time scales (daily, monthly or annual). In the model created as an example for the Piracicaba, Capivari and Jundiaí (PCJ) river basins, the time scale of the data in WEAP was monthly, whereas the data on production, energy consumption and greenhouse gas emissions in LEAP had an annual scale.

When establishing a connection between C-WEAP and C-LEAP for modeling the AEA nexus, the following steps in Frame 1 should be followed.

Frame 1. Steps for establishing a connection between C-WEAP and C-LEAP through the WEAP-KIB-LEAP *framework*

1.	definition of module ports (Ex.: InputFromWEAP , OutputToLEAP);
2.	adding module ports (addInput , addInput);
3.	definition of data transformation ports (Ex.: WEAP_LEAP_Transformation);
4.	adding transformation (addTransformation);
5.	definition of the coupling between the module ports and the transformation (Coupling c1 and Coupling c2);
6.	addition of coupling (addCoupling).

Figure 1 presents the general structure of a connection between C-WEAP and C-LEAP through the WEAP-KIB-LEAP *framework*. Each part of the algorithm is related to steps (1 to 6) to be followed.

Figure 1. General structure for a connection between CWEAP and CLEAP via the 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);
```

The names WEAP_PROJECT_NAME and LEAP_PROJECT_NAME (Figure 1) refer to the names of the models created in WEAP and LEAP, respectively, from which the data in question are intended to be obtained. For the historical conditions in the example of the application of the methodology in the PCJ basins, the projects in WEAP and LEAP received the same name, which was named the “BAU scenario”. For future projections, the LEAP project was named “Cenario RCP”, and the WEAP project was named “Cenarios RCP4.5 and RCP8.5”.

The *ComponentTypes* (Figure 1) refer to the type of entity/component of the system. The possibilities for WEAP are presented in Frame 2.

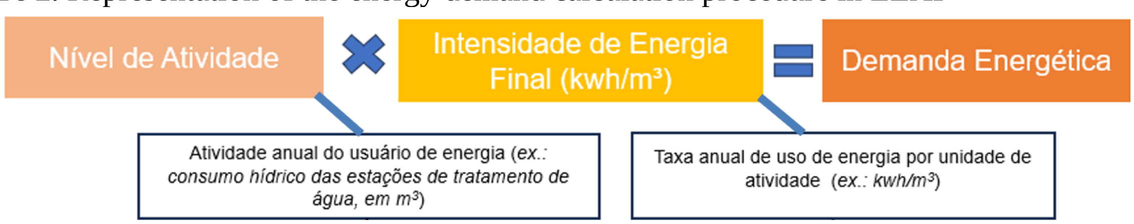
Frame 2. WEAP Entity/Component Types (*ComponentTypes)*

Node	*DemandSite* , *Groundwater* , *Reservoir* (Reservoir), *OtherSupplies* (Other water sources), *WastewaterTreatment* (Sewage Treatment Plant) and *Catchment* (Watershed Subbasin);
Link	*Transmissions* (Transmission Link), *Runoffs* and *ReturnFlows* (Return flow)
Flow	*Rivers* and *Diversions*

In LEAP, *component types* can be type resources, transformations (transformations), processes, demands or effects. The term "resource" refers to the source used in energy generation (hydro, solar, wind, biomass, etc.). The transformation is linked to the type of energy to be generated; in the case of the PCJ basins, the type of energy modeled was electrical energy and the processes involved (characteristics of the power generation plants). The demand refers to energy consumption, and the effect is the emission of pollutants.

Before proceeding, it is important to understand the procedure for calculating energy demand via the LEAP program to establish relationships between the variables. The calculation involves multiplying the factors "demand activity level" and "final energy intensity" (Figure 2). For example, assuming that in 2014, the activity level of the water treatment plants in the river basin was 1,000,000 m3, with a final energy intensity equal to 0.3 kWh/m3, the energy demand of these water treatment plants in 2014 would be 300,000 kWh.

Figure 2. Representation of the energy demand calculation procedure in LEAP



In *ComponentName* (Figure 1), the name of the component is defined. In *PortTypes, the type of port* is defined, which can be input (*Input*) or output (*Output*) data of the model created in WEAP or LEAP. In *VariableName*, the name of the variables with which you want to make the connection between the WEAP and LEAP programs is defined.

Finally, in *ScenarioName* (Figure 1), the name of the scenario is defined. WEAP, like LEAP, allows the creation of scenarios on the basis of assumptions; however, both have predefined scenarios that cannot be deleted. WEAPs are called *Current Accounts*, which refer to the first year of the historical series, and *Reference*, which encompasses the remaining years of the series. In LEAP, the predefined scenarios are *the current accounts and the baseline*.

To run the MI code, in addition to installing the WEAP and LEAP programs, it is also necessary to install the following programs: NodeJS, Visual Studio Code, Java 8, Java (latest version released), Python 2.7 and Microsoft Visual C++. Additionally, it is essential to install the following extensions in Visual Studio Code: C/C++, Intellicode API, Extension Pack for Java; JavaScript (ESG) code Snippets; Node.js Exec; Node.js code Snippets; Prettier-code formatter; TSLint Snippets; TSLint Vue; TSLint Vue -TSX; and TypeScript + Webpack Problem.

The interactions established through MI in the PCJ basins (Figure 3) were as follows:

- The volume of water demanded in the PCJ basins for urban and rural supply, in cubic meters (m^3), calculated in WEAP, was used as input data for the activity level in LEAP regarding the demand from water treatment plants (ETAs). ;
- The volume of sewage treated in sewage treatment plants (STPs) in m3, input data in WEAP, was used as input data for the activity level in LEAP regarding STP demand;
- The volume of water demanded by surface irrigation, in m3, calculated in WEAP was used as input data for the activity level in LEAP regarding surface irrigation demand;
- The volume of water demanded by conventional sprinkler irrigation, in m3, calculated in WEAP, was used as input data for the activity level in LEAP with respect to conventional sprinkler demand;
- The volume of water demanded by irrigation by a self-propelled system, in m3, calculated in WEAP, was used as input data for the activity level in LEAP regarding self-propelled demand;
- The volume of water demanded by drip irrigation, in m3, calculated in WEAP, was used as input data for the activity level in LEAP regarding drip demand;
- The area irrigated by microsprinklers, in hectares, as input data in WEAP, was used as input data for the activity level in LEAP regarding microsprinkler demand;
- Sugarcane production in the river basin, in tons, input data in WEAP, was used as input data for the activity level in LEAP regarding the demand for harvesters;
- The area planted with soybeans in the watershed, in hectares, as input data in WEAP, was used as input data for the activity level in LEAP regarding the demands: soybean harrowing, a soybean scarifier, a soybean seeder, soybean spraying and soybean harvesting. In this case, the energy consumption (diesel) for soybean cultivation was evaluated, considering different soil preparation techniques prior to planting:

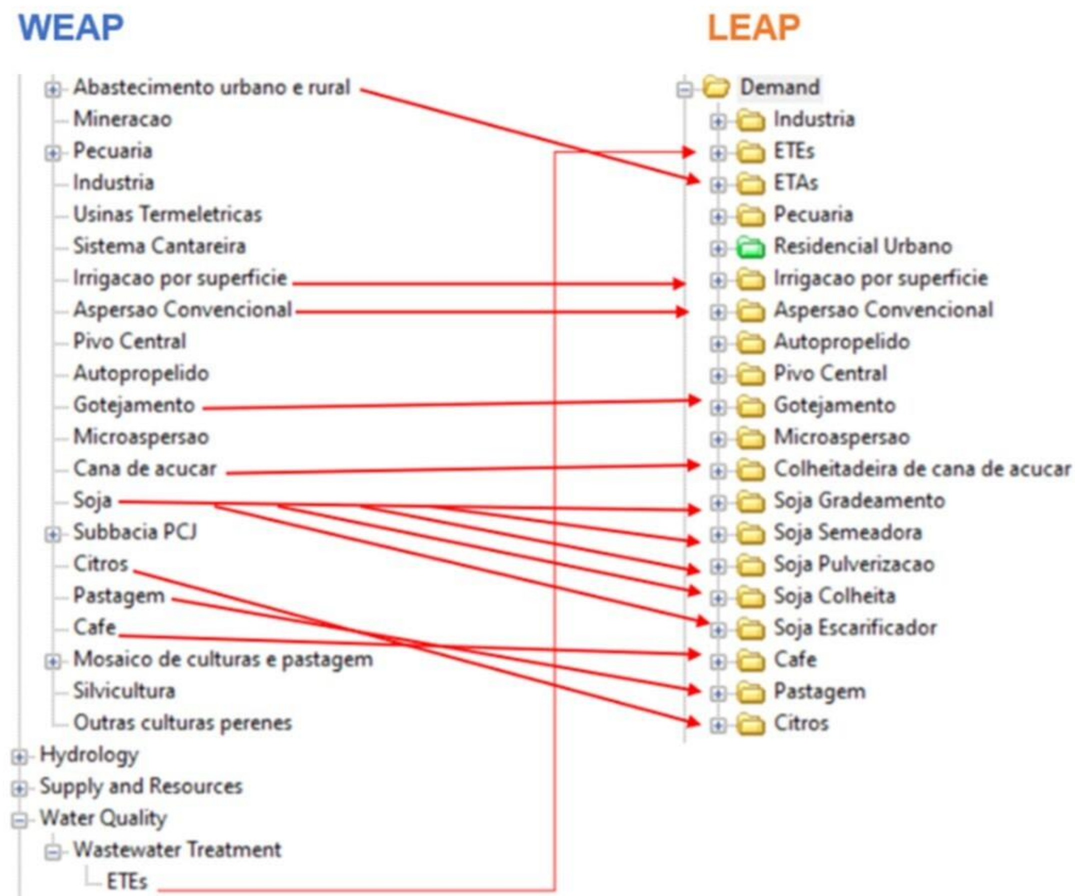
a) Minimum cultivation, a technique in which a scarifier (scarifier soybean) is used before planting, followed by the use of a seeder (seeding soybean);
b) Direct planting, a technique in which only a seeder is used before planting (soybean seeder); and
c) Conventional cultivation is a technique in which a harrow is used before planting (harboring soybean), followed by the use of a seeder (seeder soybean).

Furthermore, the energy demands during spraying (soybean spraying) and harvesting (soybean harvesting) were added.

- The area planted with coffee in the river basin, in hectares, as input data in WEAP, was used as input data for the activity level in LEAP regarding coffee demand. In this case, energy consumption (diesel) by machinery was evaluated considering the entire crop cycle.
- The area planted with citrus in the hydrographic basin, in hectares, input data in WEAP, were used as input data for the activity level in LEAP regarding citrus demand, considering all energy consumption (diesel) of the machinery used during crop management.
- The area planted with pasture in the watershed, in hectares, as input data in

WEAP, was used as input data for the activity level in LEAP regarding pasture demand, including the energy consumption (diesel) of the machinery used during pasture management.

Figure 3. Interactions established between WEAP and LEAP



Source: Silva (2023)

Importantly, both the water treated in the WTPs and the wastewater from the WWTPs are water sources with the potential to be used in irrigation.

5 RESULTS AND DISCUSSION

The MI for quantitative modeling of the AEA nexus, which uses the WEAP, LEAP and WEAP-KIB-LEAP *framework programs*, is presented below. Notably, in the WEAP and LEAP projects, the names for the elements inserted were maintained, both in historical and projected conditions. Demand projections and changes in land use and land cover were made on the basis of historical data.

```
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.LeapParameters;
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,
"Irrigacao por superficie" , PortTypes.Output, "Water Demand", "Reference"));

OutputToLEAP out2 = new OutputToLEAP("out2", new
LeapParameters(LEAP_PROJECT_NAME,
kib.few.leap.utilities.enums.ComponentTypes.Demand, "Irrigacao 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);
```

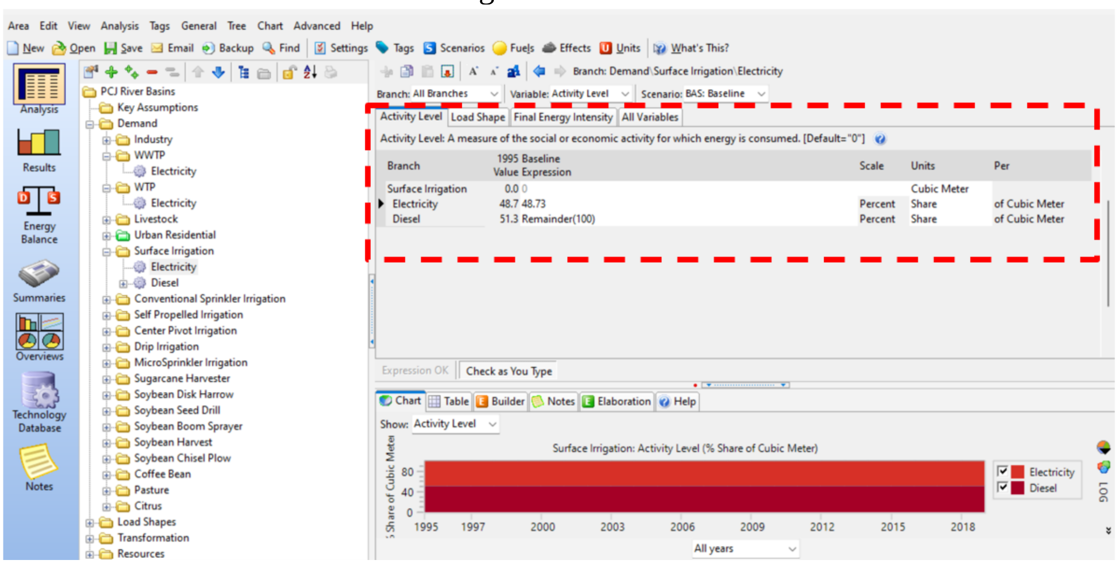
```
WEAP_LEAP_Transformation tran14 = new WEAP_LEAP_Transformation("W-LTran14");
addTransformation(tran14);
Coupling c27 = new Coupling(in14, tran14.getInput("in1"));
Coupling c28 = new Coupling(tran14.getOutput("out1"), out14);

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);
```

}
}

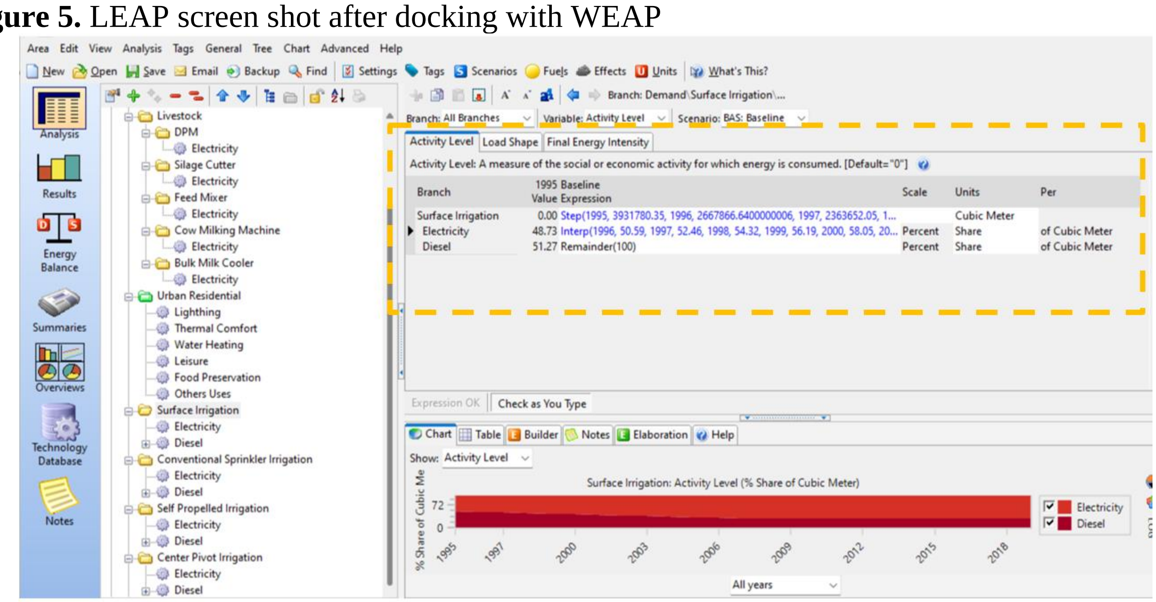
Figures 4 and 5 present screenshots showing the state of the LEAP model before and after its coupling to WEAP via MI, in the version in which the variables are translated into English. Before this process, there were no data for the activity level (*activity level*) of the energy demand for surface irrigation *in the* LEAP model. After running the MI, the mentioned field was populated with data transferred from WEAP to LEAP.

Figure 4. LEAP screen shot before docking with WEAP



Source: Silva (2023)

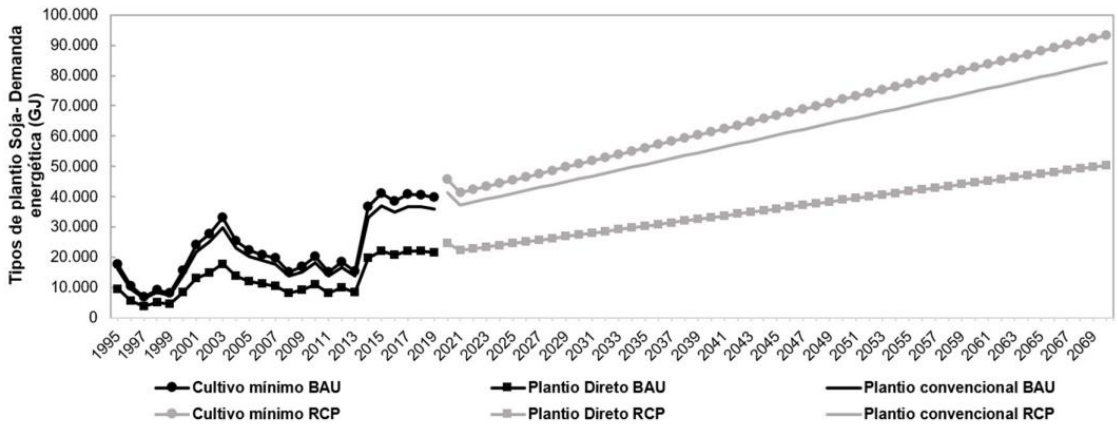
Figure 5. LEAP screen shot after docking with WEAP



Source: Silva (2023)

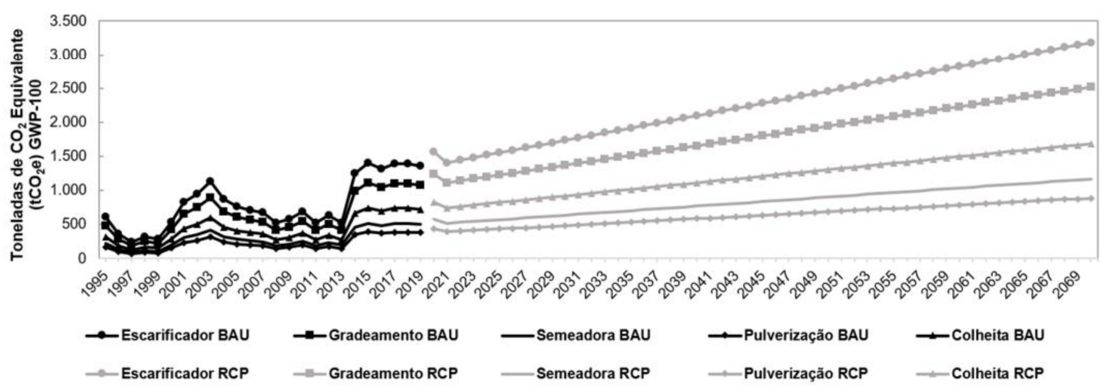
Some quantifications of the AEA nexus in the PCJ basins under historical conditions and future projections can be seen in Figures 6, 7, 8, 9 and 10.

Figure 6. Energy demand, in gigajoules (GJ), for the use of diesel in each type of soybean plant under the BAU and RCP scenarios



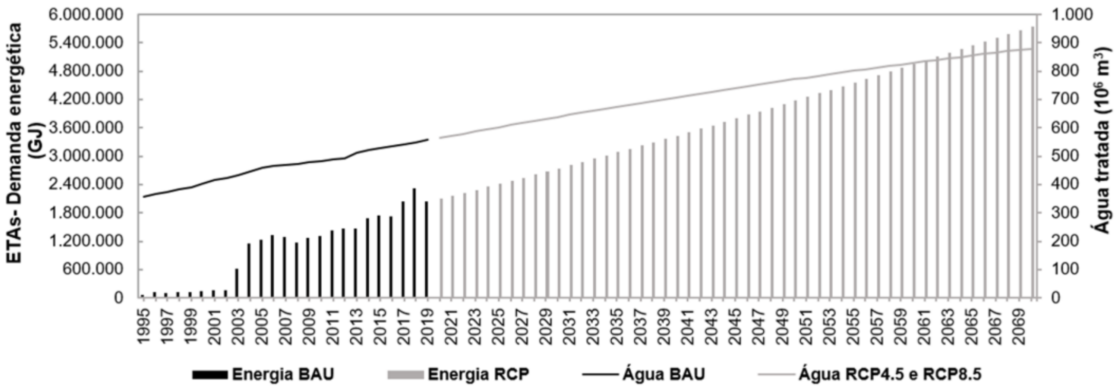
Source: Silva (2023)

Figure 7. Tons of CO2 $_{equivalent}$ (GWP-100) resulting from the type of soybean planted in the BAU and RCP scenarios



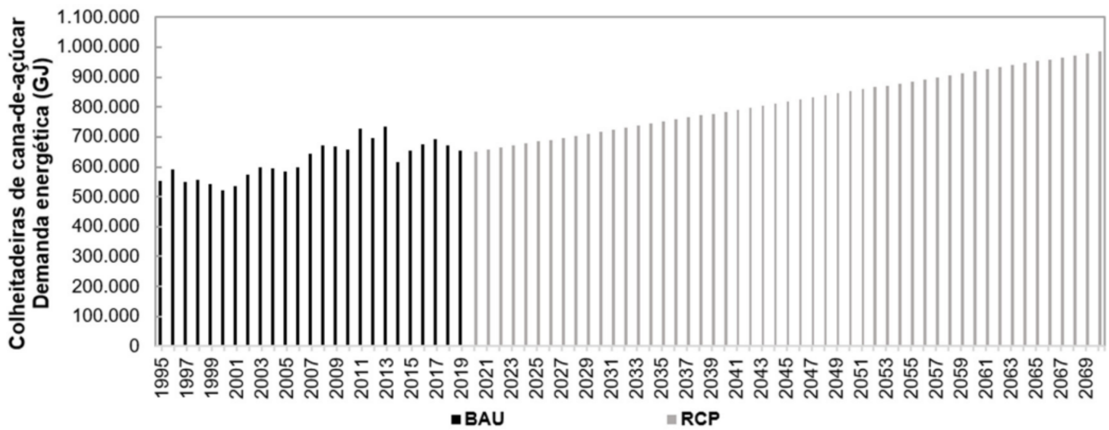
Source: Silva (2023)

Figure 8. Volumes of treated water and energy demand of water treatment plants (WTPs) located in the PCJ basins under the BAU and RCP scenarios (LEAP model) and the BAU and RCP4.5 and RCP8.5 scenarios (WEAP model)



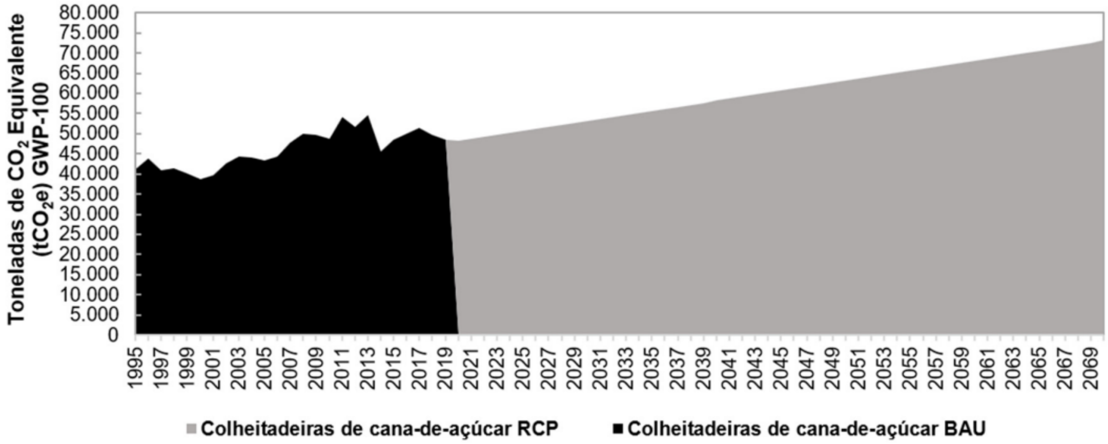
Source: Silva (2023)

Figure 9. Energy demand, in gigajoules (GJ), for the use of diesel in sugarcane harvesters in the BAU and RCP scenarios



Source: Silva (2023)

Figure 10. Tons of CO2 equivalent (GWP-100) resulting from the use of diesel in sugarcane harvesters in the BAU and RCP scenarios



Source: Silva (2023)

6 CONCLUSION

The WEAP-KIB-LEAP *framework* proved to be an effective tool for quantitatively modeling the water-energy-food nexus in river basins, allowing the integration of models developed in WEAP and LEAP with different temporal scales.

The application of the methodology in the Piracicaba, Capivari and Jundiaí river basins, with historical and projected data, provided a broad and detailed view of the interactions between water, energy and food resources. This approach offers a solid basis for planning and optimizing the use of water and energy resources, aligns with the Sustainable Development Goals, and represents a relevant contribution to the sustainable management of river basins.

It is recommended that the WEAP-KIB-LEAP *framework be applied* in different contexts and regions to assess its viability and potential for adaptation to other realities.

7 ACKNOWLEDGMENTS

To CNPq (National Council for Scientific and Technological Development), for the scholarship granted through process 143899/2019-8, which allowed full dedication to research. This work was carried out with the support of the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) - Financing Code 001.

8 REFERENCES

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

SILVA. TL **modeling of the impacts of climate change on the Piracicaba, Capivari and Jundiaí river basins considering the**

water–energy–food nexus. 2023. Thesis (PhD in Irrigation and Drainage) – Faculty of Agricultural Sciences, São Paulo State University, 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, no. 1, p. 70-87, Jan. 2019. DOI: 10.1016/j.jher.2018.10.003. Available at: http://dx.doi.org/10.1016/j.jher.2018.10.003. Access on: 1 Sep 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, no. 1, p. 29583–29603, March 2024. DOI: 10.1007/s10668-024-04586-6. Available at: https://doi.org/10.1007/s10668-024-04586-6. Access in 1 set. 2024.