

Scientific Paper

## BiomaSoft: computer system for the monitoring and evaluation of food and energy production. Part II

Julio Ramiro Quevedo-Benkí and Jesús Suárez-Hernández

*Estación Experimental de Pastos y Forrajes Indio Hatuey, Universidad de Matanzas, Ministerio de Educación Superior Central España Republicana, CP 44280, Matanzas, Cuba*  
E-mail: julio.quevedo@ihatuey.cu

---

### Abstract

In order to support and facilitate the monitoring and evaluation (M&E) of the integrated food and energy production in Cuba, BiomaSoft was designed, a computer system for the M&E of integrated food and energy production in Cuban municipalities. The objective of this paper is to provide a description of the main aspects in the implementation and development of such system, for which its use cases are textually described and the specific design class diagrams for each one are provided. As a whole, the detailed aspects show the way in which the different actors should interact with the BiomaSoft system, tailor-designed for the support of M&E. Likewise, all the compiled information allowed to design and construct a robust database, capable of supporting the large quantity of information the system should manipulate; and also to reach a deep understanding of the functioning and capacities of the system, for their later display.

Keywords: database, models, development project

---

### Introduction

The utilization of renewable energy sources, with emphasis on liquid and gaseous as well as solid biofuels has currently reached high development worldwide (FAO, 2008; IEA, 2013; IRENA, 2014). In Cuba in the last five years, advances have been achieved too in the development of these biofuels (MINEM, 2014; Suárez *et al.*, 2014), in order to contribute to food security, energy sustainability and environmental sustainability at local scale.

The international project «Biomass as renewable energy source for Cuban rural areas (BIOMAS-CUBA)», funded by the Swiss Development and Cooperation Agency (SDC) and led by the Research Station Indio Hatuey, in six Cuban provinces, contributes to this purpose. This project develops experiences for the integrated and sustainable food and energy production that needs to be monitored and evaluated, which can be favored by a software program that facilitates and automatizes all the complex process of information management.

With that objective diverse computer systems for project monitoring, control and/or evaluation were studied, such as the ones developed by

Pellerin *et al.* (2013), Acebes *et al.* (2014) and Hazir (2015), but they are focused on computerization projects and designed for developed countries. For such reason, the authors decided to tailor-design, develop and implement BiomaSoft, a computer system to support the M&E of the integrated food and energy production, which does not have antecedents in Cuba.

The goal of this paper is to provide a description of the software development process that took place in the implementation of the system<sup>1</sup>.

### Methodology

#### Description of the BiomaSoft system for M&E

From the functional and non-functional requirements, the use case diagram of the system, the actors and its general description, and following the work flow of the Rational Unified Process (RUP) methodology, provided and described in part I (Quevedo and Suárez, 2015), all the functionality of BiomaSoft is displayed in three main use cases: «Managing system data», «Managing direction strategies» and «Managing follow-up records».

---

<sup>1</sup>Part I of this paper approached the analysis and conceptualization of the environment, the need for the system was identified, and the main aspects of its design were provided: characterization of the tools and technologies used, the models and diagrams.

Below a detailed description of each one of them, as well as their specific design class diagram, derived from the generic BiomaSoft Design Class Diagram (Quevedo and Suárez, 2015), is provided, through a scheme that responds to the methodology used. This delineation will be useful for a better understanding of the system functioning, and to establish clearly who interacts with it and how they do it.

**Use case Managing system data**

Table 1 shows the basic aspects of the textual description of the use case Managing system data. The identifier of the use case, the actors who interact with it, a brief summary of its functioning, the priority it will receive by the system, and the

conditions that should be fulfilled after the use case has been needed and satisfactorily used, are thus grouped.

The «Normal flow of events», that is, the performance of the system in response to a request by the actor or user, is shown in table 2. Thus, each action of the actor, in successive order, and the respective response of the system to each one of them are listed. A prototype of graphic interface which allows to visualize the aspect the system will have and the actions the actor could perform is also presented, in the specific section of the use case Managing system data.

Table 3 shows the section that allows to eliminate a datum from the system. Associated to the Normal flow of events, it is a continuation of the list

Table 1. Basic aspects of the use case Managing the system data

Use case	Managing system data
Actors	System administrator
Summary	The use case starts when the system administrator wants to modify certain values of some datum of an already existing project, eliminate a specific one from the list, or add a new one. The system confirms the introduced data, updates the initially shown project data list and ends the use case.
Priority	Critical
Post-conditions	The project data information was updated in the system.

Table 2. Normal flow of events

Normal flow of events	System response
Actor's action	
1. The actor accesses the functionality Managing system data.	1.1 The systems lists the stored data (mark D), besides the options to eliminate (mark C), edit (mark B) and add a datum (mark A).
2. The actor selects one of the listed data (mark L) and accesses one of its respective options. The options are the following:	
<ul style="list-style-type: none"> <li>• Eliminate datum (mark C): see section Eliminate project datum.</li> <li>• Edit datum (mark B): see section Edit project datum.</li> <li>• New datum (mark A): see section New project datum.</li> </ul>	

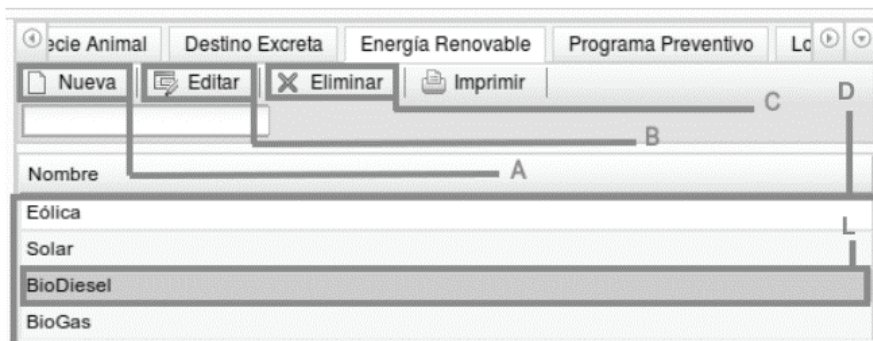
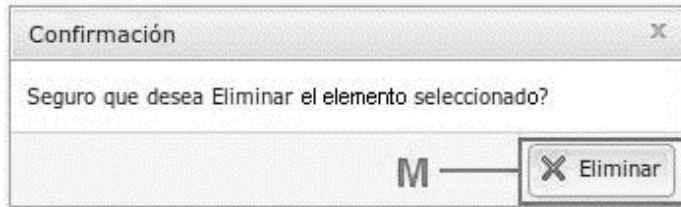


Table 3. Section Eliminate system datum

Section Eliminate system datum	
Actor's action	System response
	2.1. - The system shows an operation confirmation message.
3.-The actor confirms the elimination of the datum from the system (mark M).	3.1. - The system eliminates the datum.



of each action of the actor and the respective system response, after the actor previously accessed the option Eliminate datum (see table 2).

The section that allows to edit a system datum is described in table 4. It is a continuation of the list of each action of the actor and the respective system response, after the actor previously accessed the option Edit datum, as described by the normal flow of events (see table 2).

Table 5 shows the section which allows to add a new datum to the system. Associated to the Normal flow of events, it is a continuation of the list of each action of the actor and the respective system response, after the actor previously accessed the option New datum (see table 2).

In case that during the Normal flow of events of these above-described sections the system verifies

an error in the introduced data, an alternative flow is taken up to prevent damage in the manipulated data and report to the user. Table 6 describes an alternative flow for edition or addition, in the face of an unexpected event.

### Design class diagram

The design class diagram (fig. 1) is based on the one proposed by Quevedo and Suárez (2015), but focused on the use case Managing system data –a similar diagram is shown for the other two use cases–. This diagram allows to organize, as proposed by the architectural pattern Model View Controller (MVC), the classes and components used in such use case. This facilitates structuring the components, according to Reenskaug *et al.* (1995), Jacobson *et al.* (1999) and ISO/IEC (2014), as shown below:

Table 4. Section Edit system datum

Section Edit system datum	
Actor's action	System response
	The system shows a basis form (mark N) for the modification of the system datum information.

The actor introduces the desired changes to the system datum information in the basis form (mark N) and accesses the option Save (mark O).

The system verifies the introduced changes. The system saves the introduced changes.



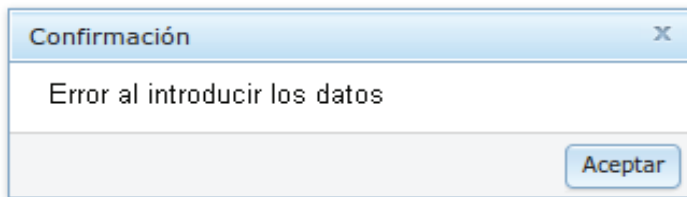
Table 5. Section Add new system datum.

Section New system datum	
Actor's action	System response
	2.1. The system shows a basis form (mark N) for the introduction of the new datum information.
3. The actor introduces the information and selects the option Save (mark O)	3.1 The system verifies the introduced data. 3.2 The system saves the introduced data.



Table 6. Alternative flow for the edition or addition of a system datum

Alternative flow	
Actor's action	System response
	3.2(a). If the data introduced are incorrect the system cancels the operation and shows a message reporting it.



Model: composed by the ORM (Object Relational Mapping) Doctrine, used by the framework (Symfony) for data management and processing.

Controller: composed by a frontal controller *GestionarProyecto.php*, in charge of receiving requests (generally URL) and, using the Symfony components (package that encapsulates the framework functioning), determining the module Actions and the action it should invoke, in order to respond to the received request.

View: composed by files with suffix success (*ListarDatosSuccess*, *NuevoDatoSuccess*, and *EditarDatoSuccess*), in charge, along with the Layout, of adhering the results of the actions, later used by the frontal controller to construct the client pages *GestionarDatosIndex*, which are the ones that will be ultimately shown to the user.

### Use case Managing direction strategies

Textual description of the use case

Table 7 shows a textual description of the use case Managing direction strategies. The use case

identifier, the actors who interact with it, a brief summary of its functionality, the priority it will receive by the system, and the conditions that should be fulfilled when the use case has ended, are grouped.

The normal flow of events is shown in table 8; thus, each action of the actor, and the respective system response to each of them, is listed consecutively. A prototype of graphic interface which visualizes the aspect the system will have and the actions the actor could perform are also provided, specifically in the use case Managing direction strategies.

The section which allows to eliminate a direction strategy is described in table 9. It is a continuation of the list of each action of the actor and the respective system response, after the actor previously accessed the option eliminate strategy, as described by the normal flow of events (table 8).

Table 10 shows the section that allows to edit a direction strategy. Associated to the normal flow of

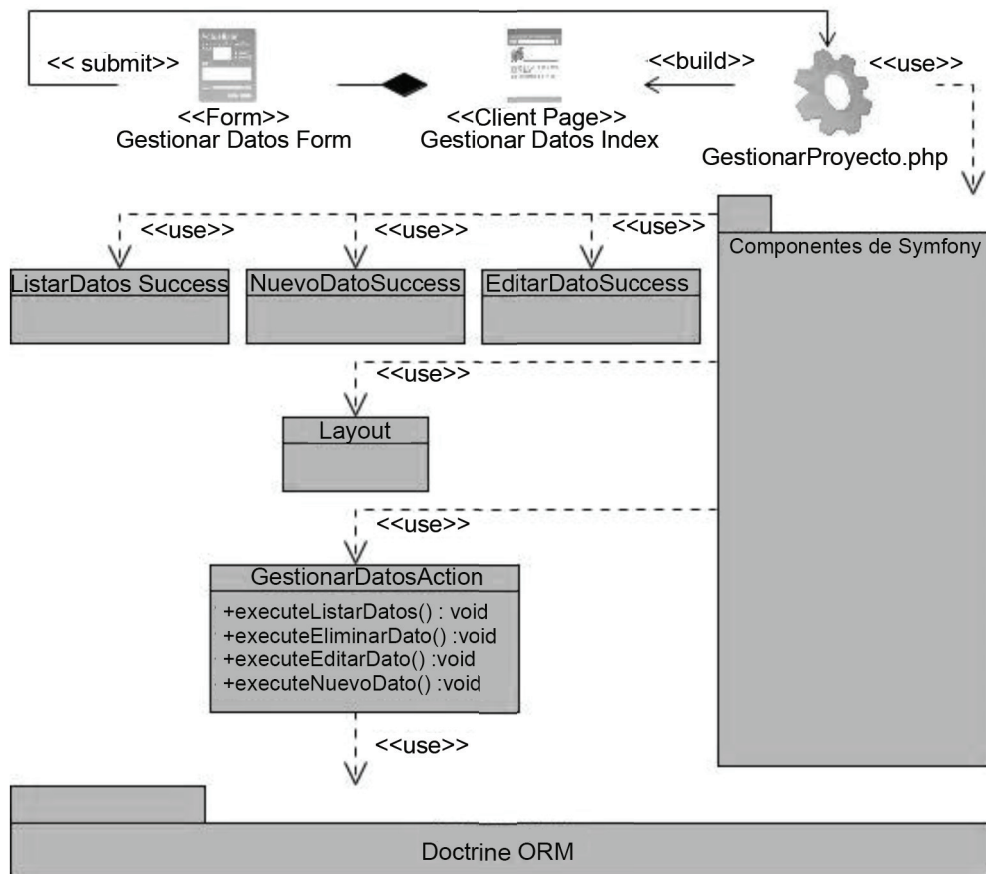


Figure 1. Design Class Diagram (Managing system data).

Table 7. Basic aspects of the use case Managing direction strategies

Use case	Managing direction strategies.
Actors	System administrator and project leader.
Summary	The use case starts when the actor wants to modify certain values of some already existing direction strategy, eliminate a specific one of the list, or add a new one. The system confirms the introduced data, updates the initially shown list of strategies and ends the use case.
Priority	Critical
Post-conditions	The information of the direction strategies in the system was updated.

events, it is a continuation of the list of each action of the actor and the respective system response, after the actor accessed the option Edit strategy (see table 8).

The section that allows to add a new direction strategy is described in table 11. It is a continuation of the list of each action of the actor and the respective response of the system, after the actor accessed the new strategy option, as described by the normal flow of events (table 8).

Table 12 describes the alternative flow for the edition or addition, in the face of an unexpected event, in case that during the normal flow of events of the above-described sections the system verifies some error, to prevent damage in the manipulated data and report to the user.

### Design class diagram

Figure 2 shows the design class diagram, which is based on the one proposed by Quevedo and

Table 8. Normal flow of events

Normal flow of events	
Actor's action	System response
The actor accesses the functionality Managing direction strategies.	The system lists the stored strategies (mark D2), besides the options to eliminate (mark C2), edit (mark B2) and add a strategy (mark A2).
The actor selects one of the listed strategies (mark L2) and accesses one of its respective options, which are the following: Eliminate strategy (mark C2): see section Eliminate direction strategy. Edit strategy (mark B2): see section Edit direction strategy. New strategy (mark A2): see section New direction strategy.	

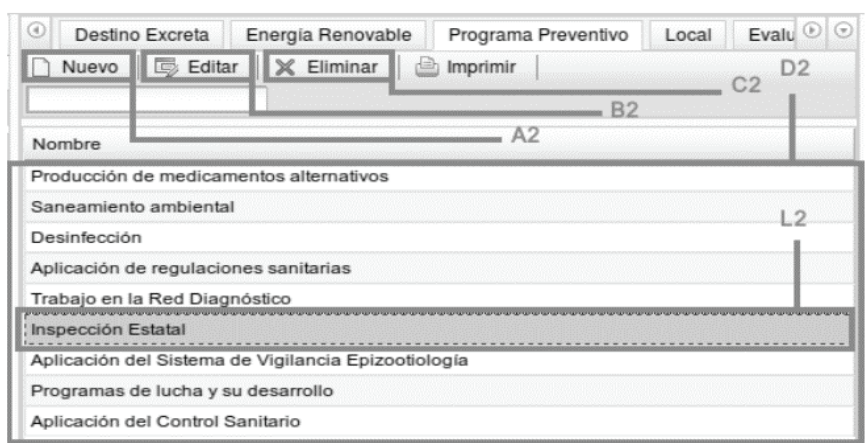
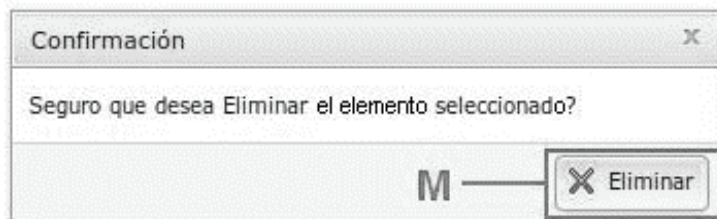


Table 9. Section Eliminate direction strategy

Section Eliminate direction strategy	
Actor's action	System response
The actor confirms the elimination of the direction strategy (mark M)	The system shows an operation confirmation message. The system eliminates the direction strategy.



Suárez (2015) and is focused on the use case Managing direction strategies.

**Use case Managing follow-up records**

Textual description of the use case

Table 13 shows the basic aspects of the textual description of the use case Managing follow-up re-

ords. The use case identifier, the actors who interact with it, a brief summary of it, the priority it will receive and the conditions that should be fulfilled after the use case has been used.

The normal flow of events describes the performance of the system in response to a request of the user (table 14). Thus, each action of the actor and



Table 10. Section Edit direction strategy

Section Edit direction strategy	
Actors' action	System response
	The system shows a basis form (mark N2), for the modification of the direction strategy information.
The actor introduces the desired changes to the direction strategy information in the basis form (mark N2) and accesses the option Save (mark O2).	The system verifies the introduced changes. The system saves the introduced changes.

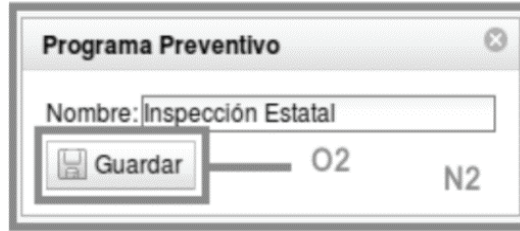


Table 11. Section Add new direction strategy

Section Add new direction strategy	
Actor's action	System response
	2.1. The system shows a basis form (mark N2) for the introduction of the new direction strategy information.
3.- The actor introduces the information and selects the option Save (mark O2).	2. The system verifies the introduced data. 3.2. The system saves the introduced data.

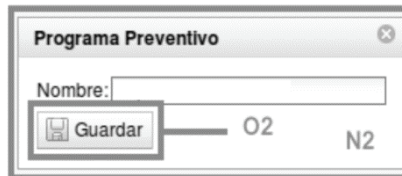
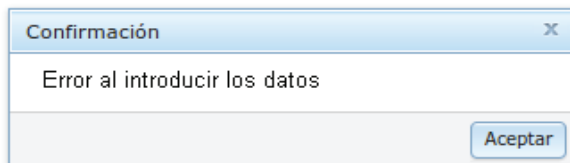


Table 12. Alternative flow for the edition or addition of a direction strategy

Alternative flow	
Actor's action	System response
	3.2(a) If the introduced data are incorrect, the system cancels the operation and shows a message reporting it.



the respective response of the system to each one of them are listed; a prototype of graphic interface which visualizes the aspect the system will have and the actions the actor could carry out in the use case Managing follow-up records, is also shown.

Table 15 shows the section that allows to eliminate a follow-up record. Linked to the normal flow of events, it is a continuation of the list of each action of the actor and the respective response of the system, after the actor accessed the option Eliminate record (table 14).

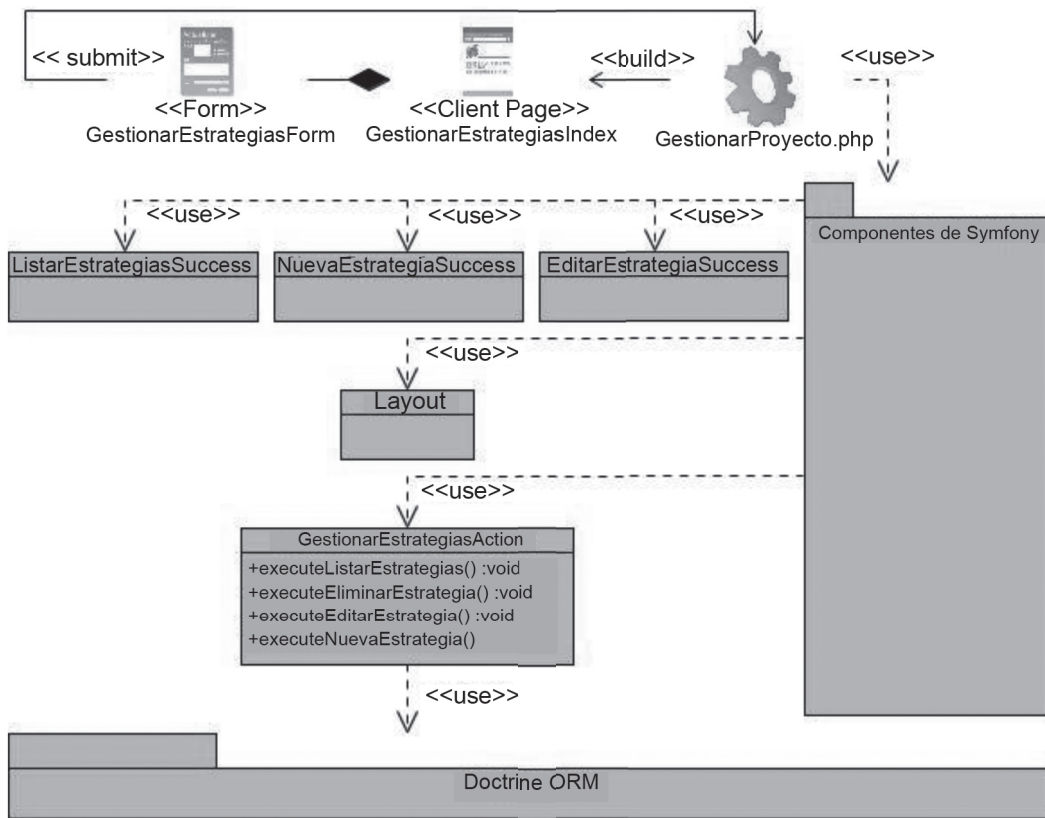


Figure 2. Design class diagram (Managing direction strategies).

Table 13. Basic aspects of the use case Managing follow-up records

Use case	Managing the follow-up records.
Actors	System administrator, project leader and person responsible for monitoring.
Summary	The use case is started when the actor wants to modify certain values of an already existing follow-up record, eliminate a specific one from the list, or add a new one. The system confirms the introduced data, updates the initially shown list and ends the use case.
Priority	Critical
Post-conditions	The information of the follow-up records was updated in the system.

The section that allows to edit a follow-up record is shown in table 16. It is a continuation of the list of each action of the actor and the respective system response, after the option Edit record described in the normal flow of events (table 14) was accessed.

The section that allows to add a new follow-up record (table 17) is associated to the normal flow of events and is a continuation of the list of each action of the actor and the respective system response, after the actor accessed the option New record (see table 14).

In case that during the normal flow of events of these sections the system verified some error, an al-

ternative flow (table 6) is taken up, to prevent damage in the manipulated data and report to the user.

### Design class diagram

The design class diagram of the system (fig. 3) is based on the one proposed by Quevedo and Suárez (2015), but focused on the use case Managing follow-up record.

### Conclusions

With the different definitions, concepts, entities and their relations, it was shown who interacts



Table 14. Normal flow of events

Normal flow of events	
Actors' action	System response
The actor accesses the functionality Managing follow-up records.	1.2.-The system lists the follow-up records (mark D3), besides the options to eliminate (mark C3), edit (mark B3) and add a record (mark A3).
The actor selects one of the listed records (mark L3) and accesses one of its respective options:	
<ul style="list-style-type: none"> <li>• Eliminate Record (mark C3): see section Eliminate follow-up record.</li> <li>• Edit Record (mark B3): see section Edit follow-up record.</li> <li>• New record (mark A3): see section New follow-up record.</li> </ul>	

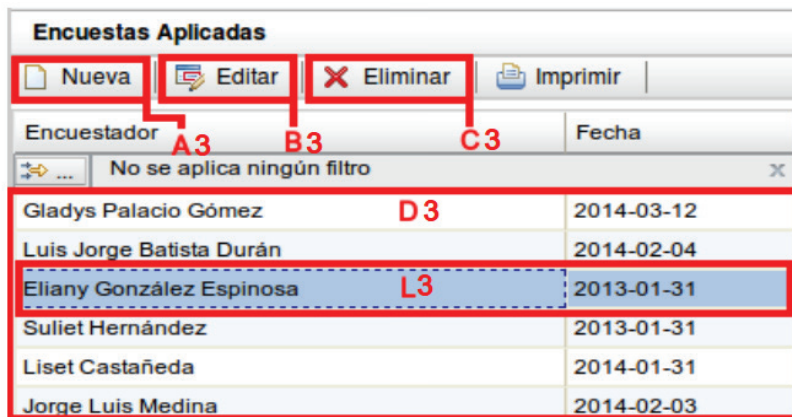
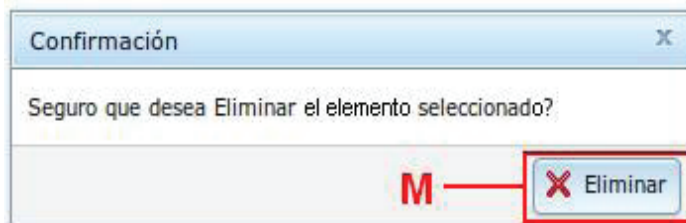


Table 15. Section Eliminate follow-up record

Section Eliminate follow-up record	
Actor's action	System response
	2.1.-The system shows an operation confirmation message.
3.-The actor confirms the elimination of the follow-up record (mark M).	3.1.-The system eliminates the follow-up record.



with BiomaSoft (project managers and monitoring and evaluation specialists at national and local level) and how they do it.

The construction, presentation and analysis of the diagrams, models and descriptions shown allowed to design and construct a robust database,

capable of supporting the large amount of information that the system must manipulate.

Likewise, a tool was obtained to support the monitoring and evaluation of the integrated food and energy production; and a better understanding of the physical and logical distribution of such production

Table 16. Section Edit follow-up record

Section Edit follow-up record	
Actor's action	System response
	2.1.-The system shows a basis form (mark N3), for the modification of the follow-up record information.
3.- The actor introduces the desired changes to the follow-up record information in the basis form (mark N3) and accesses the option Save (mark O3).	3.1.-The system verifies the introduced changes. 3.2.-The system saves the introduced changes.

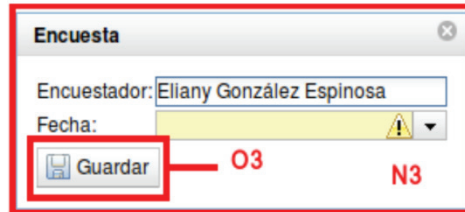


Table 17. Section Add new follow-up record

Section Add new follow-up record	
Actor's action	System response
	2.1.-The system shows a basis form (mark N3) for the introduction of the new follow-up record information.
3.-The actor introduces the information and selects the option Save (mark O3).	3.1.-The system verifies the introduced data. 3.2.-The system saves the introduced data.

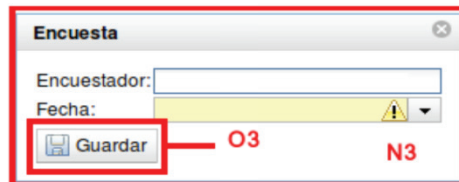
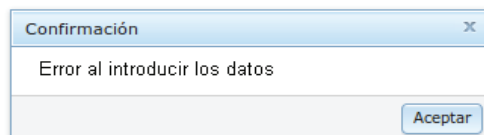


Table 18. Alternative flow for the addition or addition of a follow-up record

Alternative flow	
Actor's action	System response
	3.2(a) If the introduced data are incorrect, the system shows a message reporting it.



was achieved, which allowed to reach a deep understanding of its functioning and capacities, for its later display.

**Bibliographic References**

Acebes, F.; Pajares, J.; Galán, J. M. & Lopez-Paredes, A. A new approach for project control under un-

certainty. Going back to the basics. *Int. J. Project Manage.* 32 (4):423-434, 2014.

Allué, A.; Domínguez, E.; López, A. & Zapata, María A. QRP: a CMMI appraisal tool for project quality management. *Procedia Technol.* 9:664-669, 2013.

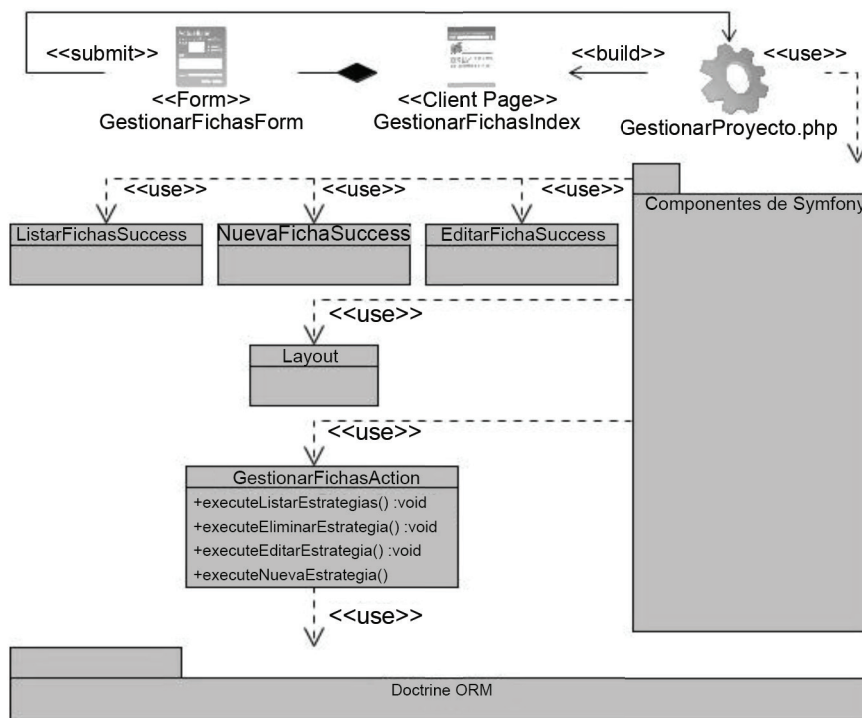


Figure 3. Design class diagram (Managing follow-up records)

Cubadebate. *Cuba apuesta por una energía más limpia, diversa y eficiente*. La Habana. <http://www.cubadebate.cu/especiales/2014/08/14/cuba-apuesta-por-una-energia-mas-limpia-diversa-y-eficiente/#.Wc5bGXop7IU>, 2014.

FAO. Cambio climático, bioenergía y seguridad alimentaria: opciones para las instancias decisorias de políticas identificadas por las reuniones de expertos. *Conferencia de alto nivel sobre la seguridad alimentaria mundial: Los desafíos del cambio climático y la bioenergía*. Roma: FAO. p. 42, 2008b.

FAO. *Política bioenergética, mercados y comercio, y seguridad alimentaria y perspectivas mundiales de la seguridad alimentaria y de los combustibles*. Roma: FAO. <http://www.fao.org/foodclimate>, 2008a.

Hazir, O. A review of analytical models, approaches and decision support tools in project monitoring and control. *Int. J. Project Manage.* 33 (4):808-815, 2015.

IEA. *Renewable energy. Market trends and projections to 2018*. Paris: International Energy Agency, 2013.

IRENA. *Global bioenergy supply and demand projections for the year 2030*. Abu Dhabi, United Arab Emirates: International Renewable Energy Agency, 2014.

ISO/IEC. *Software engineering. Software product quality. Requirements and evaluation (SQuaRE). Guide*. ISO/IEC 25000:2014. Genève, Switzerland: International Standards Organization, 2014.

Jacobson, I.; Booch, G. & Rumbaugh, J. *The unified software development process*. Reading, Mass, USA: Addison-Wesley Professional, 1999.

MINEM. Política para el desarrollo perspectivo de las fuentes renovables y el uso eficiente de la energía. *XII Seminario Nacional de Energía en apoyo a la toma de decisiones*. La Habana: Dirección de Energía Renovable, Ministerio de Energía y Minas. p. 36, 2014.

Pellerin, R.; Perrier, Nathalie; Guillot, X. & Léger, P. M. Project management software utilization and project performance. *Procedia Technol.* 9:857-866, 2013.

Quevedo, J. R. & Suárez, J. BiomaSoft: Sistema informático para el monitoreo y evaluación de la producción de alimentos y energía. Parte I. *Pastos y Forrajes*. 38 (3):209-215, 2015.

Reenskaug, T.; Wold, P. & Lehne, O. A. *Working with objects: the Ooram software engineering method*. Upper Saddle River, USA: Prentice Hall, 1995.

Suárez, J.; Cabeza, Evelyn & Quevedo, J. R. *Formulación de estrategias locales para la producción integrada de alimentos y energía en seis municipios cubanos. Documento interno del proyecto BIOMAS-Cuba*. Matanzas, Cuba: EEPF Indio Hatuey, 2014.

Suárez, J. & Martin, G. J., Eds. *La biomasa como fuente renovable de energía en el medio rural: La experiencia de BIOMAS-CUBA*. Matanzas, Cuba: EEPF Indio Hatuey, 2012.