A cloud decision support system for sow farms.

Author: Dídac Florensa Cazorla
Supervisors: Francesc Solsona Tehàs and Jordi Mateo Fornés

Abstract

Optimization models are being more used to tackle agro-business problems. This way, cloud computing, machine learning, big data, internet of things and optimization are key factors in the innovation of the sector. This paper proposes a cloud architecture to suggest a ranking of candidate sows to be replaced and a set of tools to help the farmers to make better strategic, tactical and operational decisions related with the structure of the sow farms. The cloud architecture automatizes the process to obtain the results, making the process transparent to the farmer. This work extends the advantages of optimization models with the potential of cloud computing. The results shows that the cloud architecture proposed helps decisions makers in real pig farms to make a better planning by obtaining the competitive advantages of using the proposed model in a usable, flexible and simple way.

Keywords: cloud computing, SAAS, sows replacement, optimization models, agrobusiness

1. Introduction

Smart agriculture is a development that emphasizes the usage of information and communication technology in real life agro-business problems. The emergent trends in the agro-industry involve the mixture of operation research, analytics, cloud computing, the internet of things or bigdata. The current researchers highlight the power and benefits of adopting and integrating this new methods and technology to the current agro-industrial processes. New technology such as the Cloud computing and the Internet of things are expected to leverage this development and increase the available amount of data stored, the amount of processed real-time data, the introduction of robots and drones and also the artificial intelligence. In [1], an inspiring review focused on smart farming and new technology is presented. There is a rapidly growing literature on this field such as [1], [2], or [3], which indicates the novelty and the increasing interest of combining the emerging technology with the agro-business context.

Cloud computing and Software-as-a-service (SAAS) eliminate the requirement for a powerful computer with an environment dedicated to solve complex mathematical models. The only thing needed is an electronic device connected to the Internet. With only a web browser, decisions makers obtain access to almost unlimited computing power, no matter which device is used (handheld device, desktop computer or laptops).

The pig market, little by little is evolving to be more and more competitive. Thus, pig farms need novel tools to support, suggest and help the adoption of operational, tactical and strategical decisions. There is overwhelming evidence that prolificity is a key factor in the long and short term performance horizon of a pig farm, see [4] and [5].

The replacement of sows is a well-known problem related to pigs prolificity. Authors in [6] described this problem as determining the optimal moment to make the replacement. There are a lot of models proposed to optimize the replacement politics. In [7] a stochastic dynamic programming model is proposed to maximize the present value of the expected annual net returns over a specified planning horizon. Furthermore, other authors presents an extension using multiple attributes to optimise sow replacement decisions, see [8]. Moreover, in [9] a hierarchical model has beeen proposed taking advantatge of the biological model of the replacement using a markov and bayesian process. More recent studies applied linear optimization to resolve the problem. A deterministic linear model is presented in [10] , and also a two-stage stochastic model is proposed in [11]. However, the practical application of this problem is limited due to the lack of applications and integration to the market, see [12].

Based on the work done by [13], where authors propose a tool to obtain the optimal replacement of sows in a specific farm, considering all the attributes that make a farm unique. In this paper, is presented an automation of this complex process. Furthermore, this research proposes a cloud service that merges the automation of this process with the flexibility, usability and the integration capability of a cloud service.

Thus, the novelty of this work is presenting a cloud service that can be adopted by classical and updated pig farmers. This way, farmers can take advantage of the cloud features to get a tool that can be fitted and integrated with all the current farm management software available. Moreover, traditional farmers that work today with excel and notebooks could have the opportunity to use this tool to improve their decisions. Besides, this work describes a multi-platform application to customize and get precise results in a simple, usable and smart way.

The main contribution of this work is putting complex mathematical models as a service of the farmers in an easy and intuitive way, to help and support the decisions related to the sow replacement.

The remainder of the paper is organized as follows, Section 2 presents the automation of the replacement sow policy. After that in Section, 3 the architecture that orchestrates the decision support system and also integrates the automation with cloud computing features is proposed. Further on, in Section 4 the results from a real case study are evaluated and discussed. Finally, Section 5 outlines the main conclusions and future work.

2. Replacement Policy Automation

Nowadays, the decision to replace a sow is done using the experience and the intuition of the farmer. The available evidence seems to suggest that a sow is eliminated using a company criteria or using the individual numerical productivity of each animal.

There is a rapidly growing trend in using a set of farm criteria to keep the farm structure through the time horizon. Thus, establishing farms limits is a priority for farmers. There are common constraints that theory suggests to evaluate such as the maxim number of cycles, the maxim number of repetitions or the maxim number of misbirth. However, real farms abound with other more flexible limits, such as using the ideal census, the number of maternity places or the low productivity to model the constraints. This way, the choice is simple, the animals that overcome the limits are directly eliminated. Thus, this situation claims that at the end the choice is technical and it is done without a proper economic analysis.

In the present study, the model proposed in [13] is used to obtain the optimal sow replacement policy that maximizes the benefits of the farms considering the current structure, the prolificty, the predictions about the productive behaviour and also the evolution of the sows. This way, the model represents the dynamics of the farm from an economic point of view.

The results of this model provide confirmatory evidence that the usage of this model is suitable to build a ranking of animals sorted using an economic indicator. Furthermore, the results can be used to simulate different scenarios to evaluate the situation of what would happen if I made this action, or compare the current farm structure with the ideal structure.

In spite of this, the usage of the model to make this analysis is complex and time-consuming. Technical skills are required to run and obtain the knowledge from the model results. Moreover, the introduction of the data to the model it is not trivial.

Figure 1 depicts all the manual process that must be done to obtain the results before making the decision.

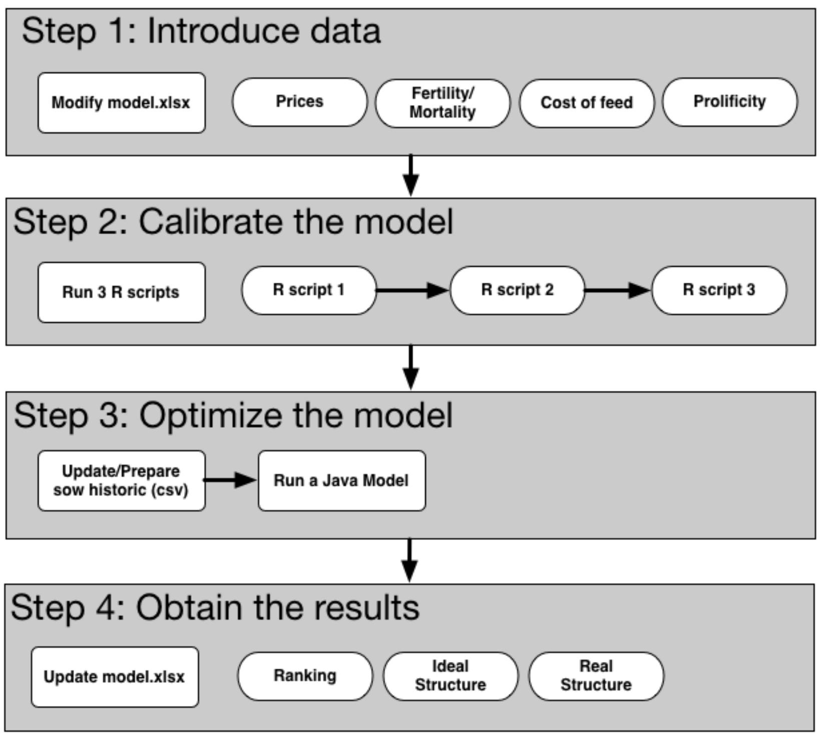

Figure 1: Manual process to make the decision of which sows must be replaced using the optimitzation model.

The first step of the process represents the stage where the farmers must introduce all the specific parameters of their farm into a prepared excel sheet template. These parameters go from the feed costs to the mortality and fertility parameters. This information is the input for the following step. Thus, the second step represents the calibration of the farm. Three different R scripts are executed using the previous information so as to generate the calibration output of the farm required by the optimizer in the following step. It is important to note that these steps are only required the first time or after a change in initial parameters. So, the farmer must keep on saving these output files. After these steps, the farmer needs to prepare a sow historical file, preferable in csv format, it can be exported from the majority of the farm management software. Then, the farmer is allowed to use the optimization model, running a Java program. This optimizer combines the historical file and the results of the calibration with a mathematical model in order to generate the optimal replacement policy. The last step represents the stage where the farmer copy the results of the optimizer to the right cells in the excel template so as to obtain the ranking, the charts with the ideal structure and other fancy results.

For the sake of discussion, I would like to argue that all this evidence is the base to justify the automation of this process. In this section, the automation of this process is presented. The authors have realized a deep research on different ways to make the automation reusable, easy to maintain and as simple as possible. In this situation authors discover the data integration tool Pentahoo Kettle. This tool fits perfect to these objectives. Furthermore, this tool allows users to merge, transform and run scripts belonging to very different kind of heterogeneous data. Besides, it has a very usable interface. The design process requires the configuration of a flow with different boxes. Each box can be configured to execute a different task or data transformation. More information about this software can be found here [14].

In spite of this, this automation is not the final solution that the farmers need to improve their economic activity since it is not completely transparent. The main problems are that this script does not avoid the installation of Java and R in the farmer computer. Moreover, another problem is the time required by the script to finalize the tasks. During all this time, the hardware resources of the farmer are going to be very busy. Furthermore, farmers must store the results in different folders if they are interested in using their historical data in the future. Thus, on the basis of this evidence, it seems fair to suggest that cloud computing can exploit this automation, avoiding these problems and making it completely transparent to the farmer.

3. Architecture

This section presents the cloud architecture designed to offer the knowledge and the optimization as a service to the pig farmer. First of all, the overall architecture is presented and analyzed. Then, the different parts that belong to this skeleton are described and discussed.

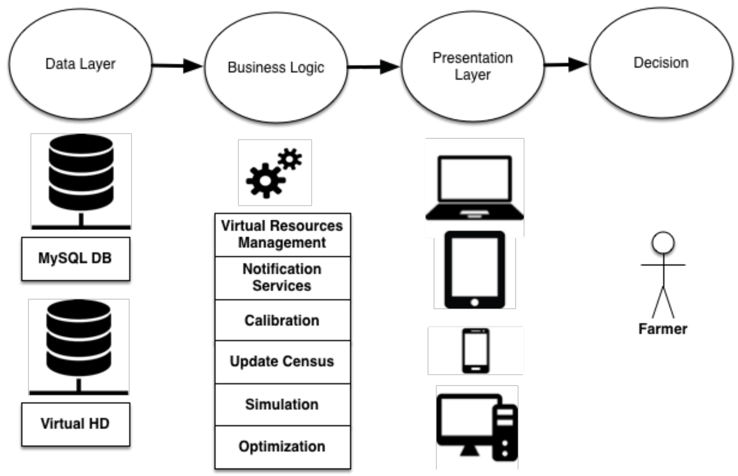

Figure 2: Architecture layer skeleton.

This architecture is built under the platform OpenNebula, see [15]. The main reason for choosing OpenNebula was to take advantage of the Stormy server. This is a private cloud deployed with the OpenNebula platform that belongs to the University of Lleida. See [16] for more information. All the parts of the cloud service were deployed on this platform using Centos7 images as the OS. The architecture is composed of three layers: the presentation layer, the business logic, and the data layer.

The main characteristic of the full system is security. The virtual machines are only accessible using the cloud service or VPN connection. The presentation layer is an exception, and it is the open door to the virtual resources. This layer acts as a middleman. Furthermore, there is a firewall that denies all connections except the ones started from the presentation layer. Thus, all the confidential data is protected inside the cloud. Moreover, the database, the private files, and the model logic is only accessible with the operations belonging to restful APIs avoiding a set of security problems.

It has been chosen this architecture because it offers some advantages very interesting to implements this app. The most important thing is that it separates the graphical interface from the data and the business logic. Figure 2 depicts the layers skeleton. This figure shows a flexible, scalable, easy to maintain and multi-purpose architecture. This way, each layer has specific

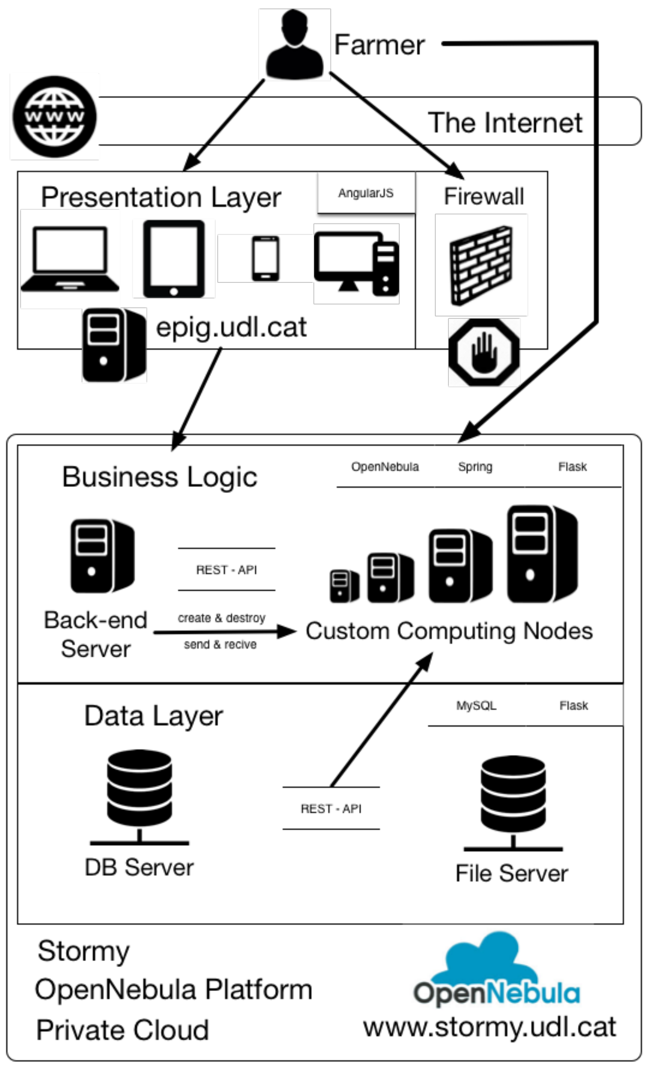

Figure 3: Architecture schema. Connection between the presentation layer, the business logic and the the data layer.

tasks and objectives. Another great advantage of this design is the capability of changing layers without perturbing the others. On the basis of this picture, it seems fair to indicate that the main function of the data layer is storing the information using different heterogeneous sources, such as a service file and a relational database. Besides, business logic is the black box and represents the optimization model, the operations with the data, the generations of reports, and so on. Finally, the main purpose of the presentation layer is the interaction with the farmer, gathering and displaying all the information.

Fig. 3 represents the overall service architecture, highlighting each of these parts and showing the main characteristics. First of all, this image indicates the way to exchange information between the layers. First of all, this image indicates the way to exchange information between the layers. This functionality is implemented using different APIs restful. Secondly, the business logic is divided into several parts. The backend server represents the operations to manage the virtual resources and the place to exchange information with the client. Further on, the computing nodes represent the virtual resources with the optimization model and all the operations that require intensive computing. Note, that these virtual resources are dynamically created and destroyed. Next, the data layer is built with two static virtual resources with a file server and a relational database. Finally, the presentation layer is a multi-platform interface that runs on the farmers' devices.

3.1. Presentation layer

The presentation layer is the visible part of the application. Furthermore, this represents the tool that the farmers see and which interacts with. The presentation layer is a web application accessible from any modern browser and any device connected to the Internet. This a soft layer that does not require too much computational resources. This way, this layer can be executed on the farmers' devices. Thus, the only requirement to obtain the reports and the analytics is a device with an active internet connection. The main purposes of this layer are allowing the farmers to register to the decision support system, upload and store their data, obtain the analytics, get access to an interactive dashboard and the last but not the least support their tactical and strategical decisions. To sum up, the user interface of the presentation layer makes the interaction between the mathematical models, the data, the analytics and the farmers as easy and usable as possible. The application is divided into 8 sections:

- **Log in**: This page is where the users logging to the application and load all the information.
- **New session**: This page is where the users create a new calibration for a concrete farm, entering the specific parameters that make a farm unique. It is explained in greater detail below.

- **Dashboard**: On this page, users can operate with and see the analytics obtained after all the calculations. This is explained in a greater detail below.
- **Charts**: This section contains the analytics. This information is useful to support the decision process.
- **Ranking table**: This page shows a dynamic and interactive table that represents the sows ranking. This is one of the most valuable results from the farmers' point of view.
- **Settings**: This section contains different views to represent all the inputs that the farmer has introduced to the system. Moreover, allows farmers to interact with them.
- **User profile**: This page contains personal data of the farmer.
- **Contact**: This page contains a form to send an email to the administrator of the app.

The main part of the presentation layer is the **Dashboard**. This is the private part of each farmer. Here, the farmers can interact with the model, launching operations, checking the results, obtaining the reports and analytics, getting or updating the farm parameters and more. One of the features of this **Dashboard** is that represents the analytics of a farm in a specific date of calibration. So, the **Dashboard** includes tools to navigate throughout all the set of farms and calibrations belonging to a farmer. Another important feature of this view is the responsive implementation that fits perfect the analytics to different screen devices. Figure 4 depicts how this screen looks like.

For the sake of discussion, I would like to argue that the view **New Session** is a smart step-by-step form that allows farmers to introduce their parameters to calibrate their farms. This way, the form is based on 6 steps. The main feature of this form is the usability. Note that, this form allows farmers to introduce the historical file that represents the current state of the animals that are currently inside the farm. Besides, other specifics parameters can be introduced using this step by step form. The first step contains the session configuration, the main parameters are the name of the farm and the type of operation. The next step represents the parameters related to the production, such as the maximum number of reproductive cycles, the range

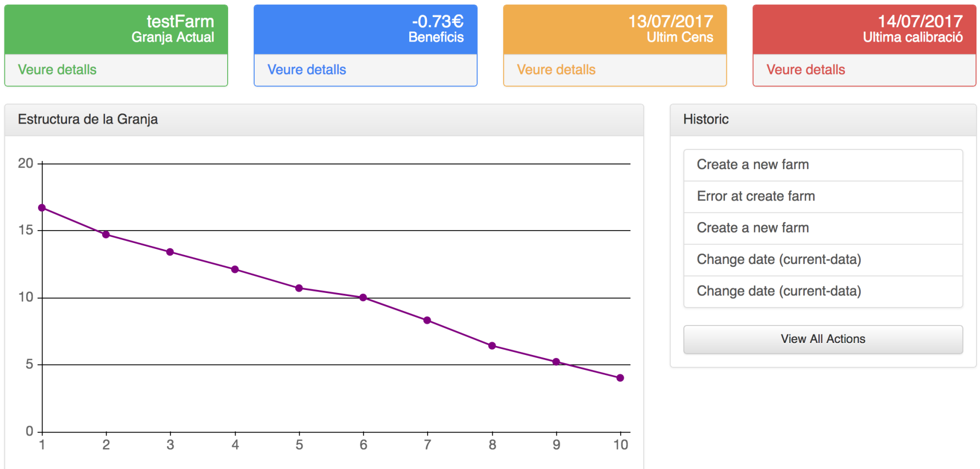

Figure 4: Dashboard view.

of coverages or the range of the weeks related with the lactation. This way, the third step is the census, so here is the place to update the historical file. Further, the fourth and the fifth steps consider the parameters related to consumption and prices respectively. Finally, the last step requires the productivity parameters, such as the mortality. In spite of the fact that there are a big set of parameters that the farmers can introduce or update, the majority of them are optional. This means, that only the historical file is required to obtain results. However, the accuracy of the solution is related to the number of parameters that the farmer introduce. As it is said in this work each farm is unique, Thus, the higher the number of parameters the model has, the more accurate will be the results. Thus, the model is going to reflect better the reality of the farm depending on the value of these parameters.

Other important analytics are the charts representing the structure of the farm. These charts can be understood as a real photo of the current state of the farm, and as the ideal structure that the farm has to be adjusted. Moreover, other crucial analytic is the chart related with prolificity. The results provide confirmatory evidence that the animal ranking is one of the most interesting tools from the farmer point of view. Thus, this view is crucial for the decision support system that this research is proposing. In this view, a dynamic and interactive table is implemented. This table contains a ranking

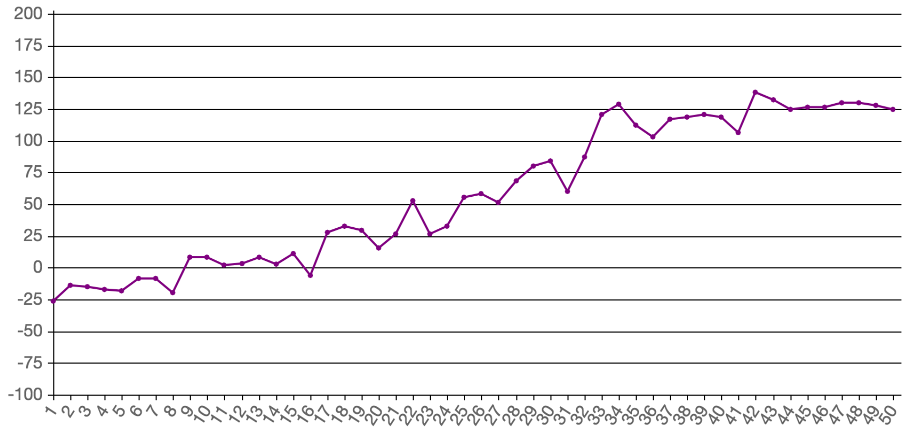

Figure 5: Example of the view from economic valuation with test data.

from the current pigs available on the farm. This ranking is based on the economic valuation extracted from the optimization model. Furthermore, the table contains different columns such as the ranking, the id of each pig, the cycle, the number of piglets in the last birth, the economic evaluation of the pig and the current state of the animal. This table allows farmers to switch the order of the ranking, reorganizing it by using other criteria. Besides, the last but not the least, this table allows filtering by key words. This mean that farmers can search quickly, specific animals, so as to improve their decisions.

Figure 5 depicts the overall economic valuation of the farm. Moreover, shows the individual economic valuation of each sow. The available evidence seems to suggest that the sows with negative or low valuation are the candidates to be replaced.

On these grounds, we can argue that Fig. 6 is crucial, because represents the same information that the chart discussed below in a way that the farmer can interact, filter, check, modify and realize operations with the data. Furthermore, Fig. 8 represents the prolificity curve and Fig. 7 the ideal structure. These are very interesting results. The main purpose of them is showing the farmer the differences between his reality and the optimal reality it could have following the suggestions of the results.

Finally, the dashboard menu, see Figure 4 allows the farmer to launch the operations. The first operation is the calibration. This operation prepares the system to be optimized quickly. The second operation is the optimization.

entries

RANKING	ID ANIMAL	CICLO	LECHONES ÚLTIMO PARTO	VALORACIÓN ECONÓMICA	ESTADO ANIMAL	DESCARTAR
1	1006	10	7	-26.2		
2	1201	10	12	-19.56		
3	1127	9	10	-18.12		
4	1079	9	11	-17.05		
5	1060	9	12	-14.88		
6	1037	9	13	-13.75		
7	1140	9	16	-8.35		
8	1173	9	16	-8.35		
9	3019	6	6	-5.97		
10	3008	6	11	2.15		

Figure 6: Ranking table view with sample data.

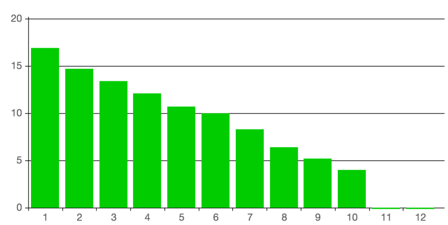

Figure 7: Ideal structure chart with sample data.

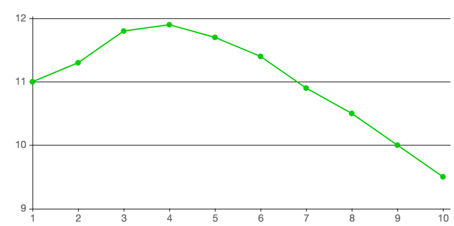

Figure 8: Prolificity curve chart with sample data

Each time that the farmer updates the information of the current state of the pigs an optimization operation must be done to ensure the robustness of the solutions. Note, that this is the reason for showing the date corresponding to the last calibration and the date corresponding to the last update. The third operation allows the farmer to update the animals state and the last operation is for generating a PDF report.

To sum up, this layer is where the data from the farmer is recollected and stored in the system. Furthermore, is the part where the farmer can launch the optimitzation models, visualize the analytics, extract the knowledge and get the suggestion and the feedback to support their tactical, operational and strategic decisions.

3.2. Business logic

The Backend server is the core of the application. It contains all the application logic and is the communication bridge between the client, the database and the different computing node. The server was implemented in Java using the application framework Spring [17], which is also based on the Model-View-Controller (MVC) pattern, and all its dependencies are managed using Maven [18].

The Backend server has to provide all the information needed by the client, so it can display it in a user friendly way, and also receive all the information provided by the user and save it or transform it as needed.

As explained above, all communication between the server and client is done through REST API calls. These calls are intercepted by the Spring controllers, where different API calls are mapped to Java methods and POST methods which can receive input and generate a response which is sent back to the client.

Moreover, all the information about sessions, calibrations, results and more need to be stored in the data layer so users can retrieve it later. This communication is done through another API rest. These GET and POST calls are implemented using Python and the Flask framework.

All the user entities stored in the database are represented in the application as Java classes, forming the application model. Communication between the application and the database uses the Hibernate framework [19], which offers a set of Java Annotations to specify how the data stored in this Java classes is mapped into database tables. Once the model entities are mapped to the database, they can be stored or retrieve from the database using the Spring repository controllers.

The virtual machines where the models (calibration and optimization) and the automation script are executed are hosted by the cloud computing service, OpenNebula. Through the API offered by OpenNebula, the application is able to communicate to obtain information about the Virtual Machines created, like their IP, which is later used to communicate individually with each Virtual Machine. Furthermore, the Backend server is responsible for creating the virtual machine when the farmer launches the operation and also to delete this machine when the operation is finished. Moreover, this server uses a queue for the tasks that can not be served due to the lack of virtual resources.

The business logic receives all the data from the presentation layer, transforms and manipulates this data and ensures that this data is stored in the

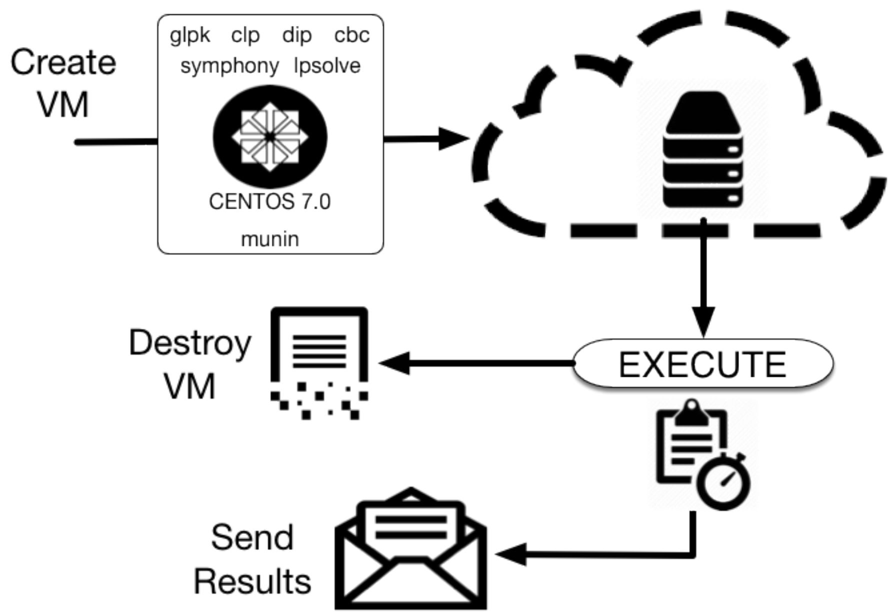

Figure 9: Life-cycle of a computing node.

data layer. On the one hand, the Backend server can be considered the brain that orchestrates all the skeleton. On the other hand, the computing nodes can be understood as the heart and the lungs, responsible for solving the hard computational tasks. The process in the computing nodes represents first of all the execution of the automation script described above and after that the notification via email to the farmer about the ending of the operation. Figure 9 represents this behaviour. It is shown how the virtual instance is created, next the work is executed, after that the farmer is notified and the virtual resource is destroyed. Thus, the main characteristic of these computing nodes is volatility. It means that computing nodes are dynamically created and destroyed, on demand.To sum up, this is the layer where the work is done. Is the main part of the system.

3.3. Data layer

The concept of a data layer is becoming more important for digital transformation. A data layer helps you collect more accurate analytics data, store persistent information about the users and more.

This service uses the data layer to store all the information related to the farmers, the farms, the animals and also the operations results. Thus, this layer is divided into two different servers. There is a relational database which stores all the data listed above. However, the historical files that represent the state of the farm and the animals in a specific format are stored in a file server with a big persistent hard disk.

The database is implemented with MySQL [20] language because is an open source software, with a good performance and lower cost to keep and run the necessary procedures. The relational schema proposed is depicted in Figure 10. This figure shows all the entities stored in the database and also their relations. The user table contains all the personal information belonging to the farmer. This way, the farm table contains the information associated with the farm, such as the dimension, breed, capacity and more. Moreover, the calibration table contains the information related to a farm and a farmer in a specific moment of time. The calibration table is the core of the information in the database. All the results corresponding to the execution of the operations need a calibration entity to be accessed. Furthermore, all the specific farm parameters need to be stored and associated with a calibration. The other entities are responsible for storing other data such as the farm operations, logs, devices used and more.

A closer look at the data indicates that this database is not enough to store the historical files. This way, a complementary file server is required. This server is based on a virtual hard disk, containing this information organized and secure. The major part of the data stored is in CSV format.

4. Case Study

This case study lies at the heart of the discussion on the usability of the service presented. The purpose of this guide is proving that expert and non-expert farmers are capable to use the proposed service. So, first of all, the steps required for making a calibration are described and after that, an optimization operation is launched. Next, the results are analyzed. The data for this study belongs to a real farm company. However, for privacy reasons, this data is been truncated.

The starting point to use the application is the login. This application uses personal data for farms and farmers, so the authentication is crucial to keep data safe. Figure 11 displays the login view. However, if the farmer is

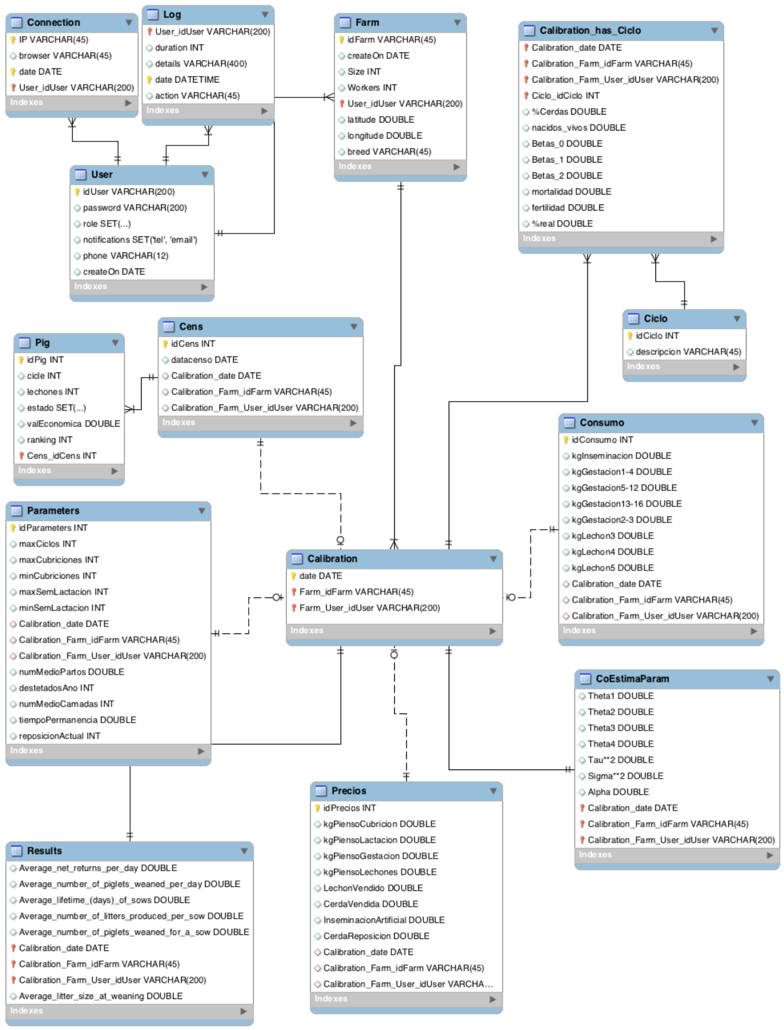

Figure 10: UML Database design.

new to the application, the starting point is the register. Figure 12 shows the simple form that allows farmers to create a profile in the application.

Once the farmer is logged in the application, he is allowed to enter into his private space. The first page displayed is the main page. This page is a command page that allows farmers to create calibrations or instead check the dashboard. Note, that on the top and on the left there are responsive menus. On the one hand, the menu on the top is responsible for the actions related to the user settings, the current session, and the main operations.

On the other hand, the menu on the left allows the navigation throughout all the sections and the access to the specific operations such as calibration, optimization or generating the report.

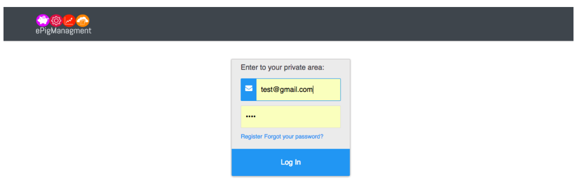

Figure 11: Login view.

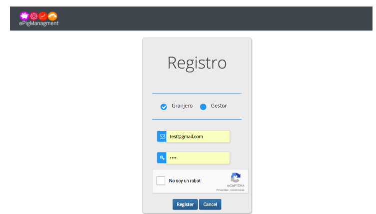

Figure 12: Register view.

Then, the first thing the farmers needs to do is realize a calibration. This way, the farmer is going to create the first farm and introduce all the required information. This action is executed by clicking the corresponding button. See Fig. 13.

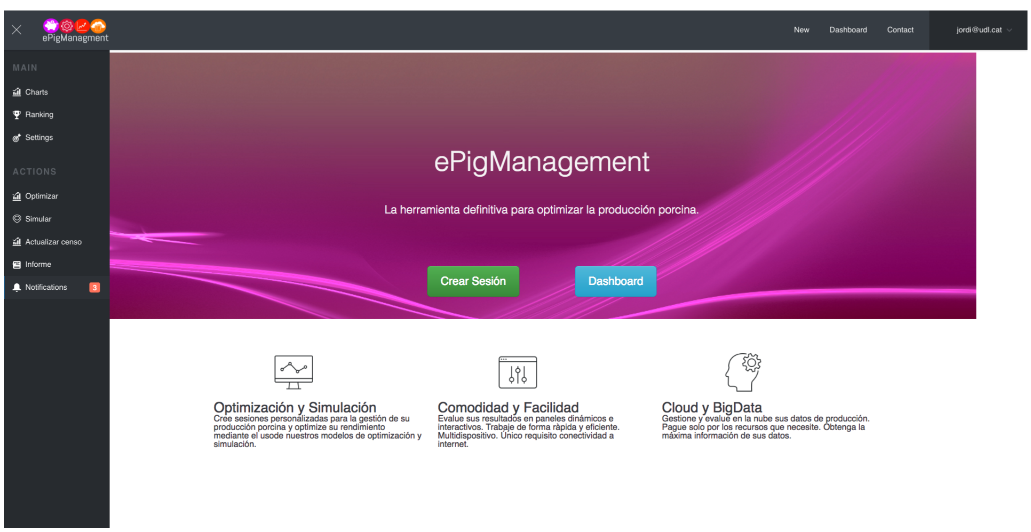

Figure 13: Main page.

Now, the farmer starts with the first step of the calibration. This step requires the creation of a farm by entering a unique name. In spite of this, if it is not the first time, the farmer is allowed to chose a farm previously created. Then, the farmer needs to choose the operation. Logically, if it is the first time, only the calibration is going to be available. Finally, the farmer has to introduce the way to receive notification. So, in this example, testFarm is going to be created and calibration and email are going to be selected. See Fig. 14.

Fig. 15 represents the next step. Here the farmers need to indicate the maximum number of cycles, the minimum and the maximum number of covers and the minimum and maximum weeks of lactation. All this information is optional, so if the farmer does not introduce any value, generic values are going to be considered. Once all the inputs are informed, click "Next". After this step, the farmer has to introduce the census information. This way, the farmer needs to upload or drag and drop the CSV file corresponding to the current census. The parameters are the current percentage of replacment, the time of permanence inside the farm, and the number of sows per cycle. See the Fig. 16. Let us introduce a set of value and click Next.

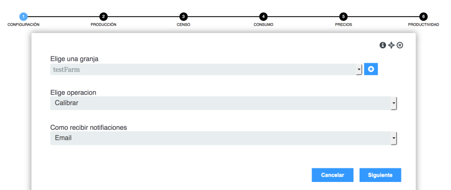

Figure 14: Step Form 1: Configuration.

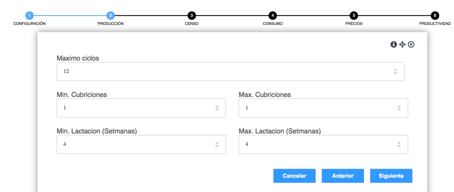

Figure 15: Step Form 2: Production

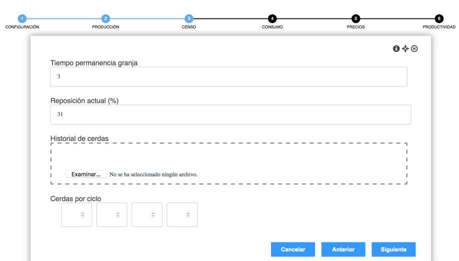

Figure 16: Step Form 3: Census.

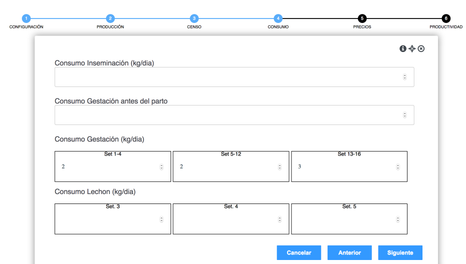

Figure 17: Step Form 4: Consumption.

The fourth step is the consumption. The farmer must inform the consumption in kilograms during the insemination and the gestation before the birth. Then, the kilograms in the gestation divided by weeks. Finally, the consumption of the piglets. Fig. 17 depicts this view and shows all the field.

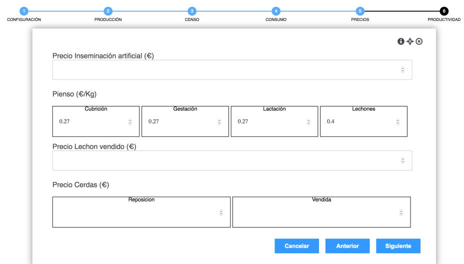

Figure 18: Step Form 5: Prices.

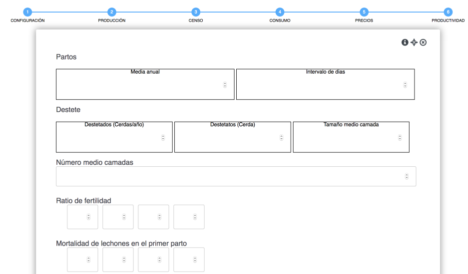

Figure 19: Step Form 6: Productivity.

The next step considers the cost corresponding to the overall process. The values required, as it can be seen in the Fig. 18 are the cost of artificial insemination, the prices of the nutrition, and the price to buy a sow and a piglet in euros.

The last but not the least, the productivity step. Here, the farmer indicates the average number of births per year. Next, the weaning information. After that, the information about fertility and mortality per cycle. This fields and the values considered for this example are shown in Fig. 19.

Once the farmer clicks Next, a page is showed informing that the inputs are properly fulfilled and also notifies that the calibration process is going to start. Let us click the button End Up and wait for the system to realize the calibration. Now, the machine state is Executing, which means that the execution has already started on the virtual machine. Only remains to wait.

When the farmer receives the notification that the calibration has finished, the farmer is allowed to check the results and analyze them. To consult this results, the app contains the Dashboard, see Fig. 4. Here, the user can obtain information related to the current structure of the farm, a historical list of the actions realized by him in the system, and also other important information related to his data. This is only a summary, to obtain more information, the farmer needs to navigate to the other views.

In the present study, the issue under scrutiny is the benefits of the decision support system proposed. Thus, a real case study is presented so as to highlight the strength and potential of the system. As it is said in the introduction of this section, the data used has been truncated. The data belongs to a real local company for the past season. Further on, the results of the calibration and the optimization of this data gathered from the real world are going to be discussed and commented. The parameters values used are the ones displayed in the user guide section.

The farm studied has an amount of 50 sows with a huge economic value. The ranking table makes a classification of these 50 sows in function of its economic value. The current economic valuation of the overall farm is around 3000 euros.

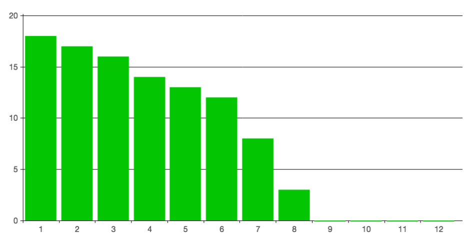

Figure 20: Case Study: Ideal structure.

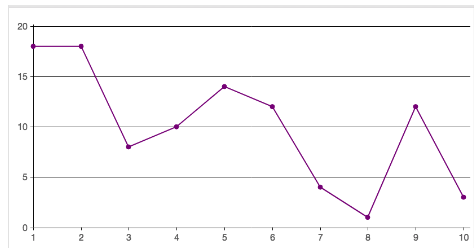

Figure 21: Case Study: Real structure

The data yielded by this optimization provide convincing evidence about the following decisions to make. The prolificity curve has been calculated using the farm's data of historical productions. This curve shows the more probable amount of the size of the litter that it could be obtained by cycle and also its trend, see 23. As it can be shown in Figure 21, the maximum amount of piglets that born alive is 12,3, this value is reached in the fourth cycle. Further on, the expected size of the litter decreases slowly until the last cycle. In the last cycle, cycle 12, on average the farm would get a litter size of 10,6 piglets. This information reflects an important part of the reality of the farm.

The next figure to analyze is the current structure of the farm, see Figure 20. This image presents an irregular curve, that is not the optimal one. The optimal curve has to decrease slowly cycle by cycle. However, a closer look at the chart indicates an extreme decrease near the 8% in the third cycle. Then, this census increases again in the order of 14% in the fifth cycle and finally decreases again till the 8^{th} cycle, where no more sows are presents.

Figure 22 represents the sows classification. Using this information the farmer makes a replacement of 5 sows. If the farmer select the more negative sows in the ranking, the farmer will obtain a overall profit around 95,81 euros. Other replacement politics will obtain less profit, indeed, some of them, could represent losses, check Figure 24.

Finally, Figure 25 represents some of the economic indicators of the farm.

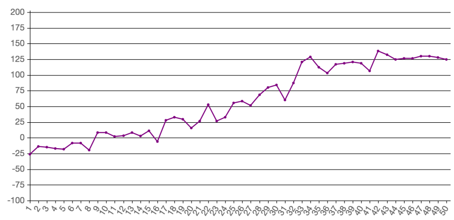

Figure 22: Case Study: Ranking.

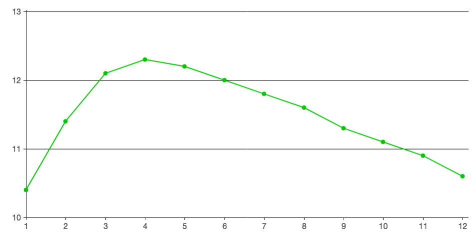

Figure 23: Case Study: Prolificity

	Resultado optimizador	Resultados granja
Cerdas repuestas	5	5
Valoración económica reposición	Ahorro de 95,81 €	Pérdida de 175,95 €
Diferencia entre valoraciones	**271,76 €**	
Valoración tras aplicar reposición	**3005,48 €**	**2733,72 €**
Valoración económica explotación	**2909,67 €**	

Figure 24: Case study: Summary results.

This information is useful to understand the advantatge of becoming similar to the ideal structure. This structure is obtained using the optimizer considering all the parameters that make the farm unique. Thus, in this case, becoming similar to the ideal structure shows a possible profit for each pig around 450,7 euros, which represents an increment around 21% of the profit for each sow.

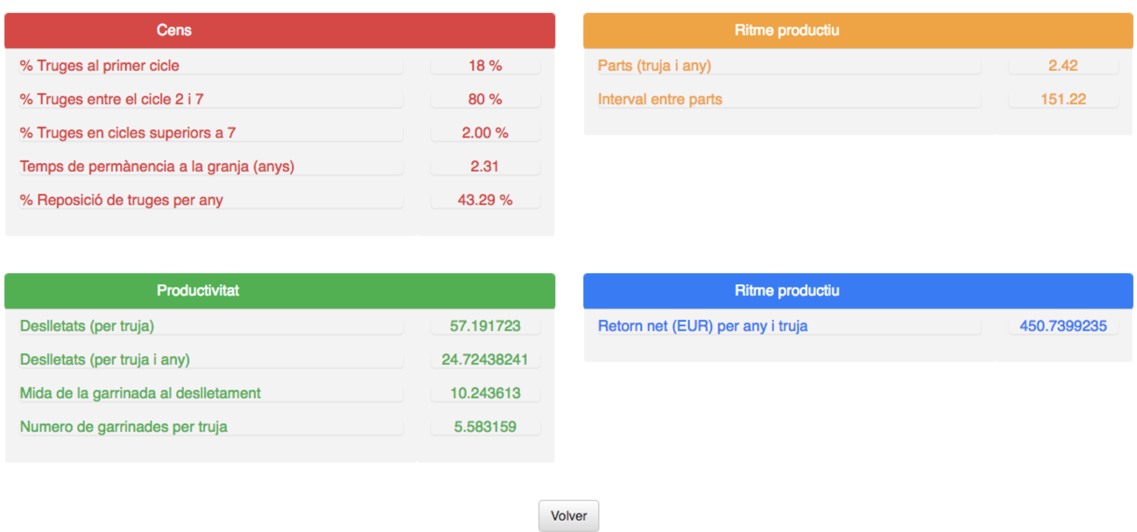

Figure 25: Case study: Indicators.

There is overwhelming evidence explained in this study to support the adoption of this system for the farm community. Furthermore, the economic benefits and the simplicity makes this system powerful and indicated to resolve real life problems and help farmers to take better strategical, tactical and operational decisions.

5. Conclusion and Future work

The novelty of this research is the development of a usable, flexible and scalable DSS to support the decision making process in the agricultural context. Moreover, the capabilities of automation and integration are the key-factors that differentiate this service from others. The service presented in this article has a lot of potential because is a web application, accessible with all devices from anywhere. Thus, the service presented in this article is a powerful seed for a much bigger service with a great potential to become a reference in the agribusiness world.

Merging the potential of cloud computing and optimisation models with usability and portability makes the farmers life easier and comfortable. This way, the process of making strategical, tactical and operational decision becomes easy.

Regarding the future work, is important to highlight the ability of the model to simulate different scenarios. Implementing this feature will allow farmers to make better strategic decisions. Less important for the decision support system but related with the usability and the comfort of the service it will be crucial the implementation of a more deep notification service. Finally, the storing and evaluation of farm historical opens the world for a non-relational database. Integrating Mongo and Spark could be a key factor. The exploitation of these data could change the pig farms reality.

6. References

[1] Sjaak Wolfert, Lan Ge, Cor Verdouw, and Marc-Jeroen Bogaardt. Big Data in Smart Farming – A review. *Agricultural Systems*, 153:69–80, may 2017.

[2] Shuwen Jiang, Tian'en Chen, and Jing Dong. *Application and Implementation of Private Cloud in Agriculture Sensory Data Platform*, pages 60–67. Springer International Publishing, Cham, 2016.

[3] Hyeon Park, Eun-Ju Lee, Dae-Heon Park, Jee-Sook Eun, and Se-Han Kim. Paas offering for the big data analysis of each individual apc. In *Information and Communication Technology Convergence (ICTC), 2016 International Conference on*, pages 30–32. IEEE, 2016.

[4] J Pomar and C Pomar. A knowledge-based decision support system to improve sow farm productivity. *Expert Systems with Applications*, 29(1):33–40, jul 2005.

[5] S Boulot, H Quesnel, and N Quiniou. Management of high prolificacy in French herds: can we alleviate side effects on piglet survival. *Advances in Pork Production*, 2008.

[6] R B M Huirne, A A Dijkhuizen, P van Beek, and J A Renkema. Dynamic programming to optimize treatment and replacement decisions at animal level. *Animal Health Economics*, chapter 7(7):85–97, 1996.

[7] R.B.M Huirne, A.A Dijkhuizen, and J.A Renkema. Economic optimization of sow replacement decisions on the personal computer by method of stochastic dynamic programming. *Livestock Production Science*, 28(4):331–347, aug 1991.

[8] R. B. M. Huirne and J. B. Hardaker. A multi-attribute utility model to optimise sow replacement decisions. *European Review of Agricultural Economics*, 25(4):488–505, jan 1998.

[9] Anders Ringgaard Kristensen and Thomas Algot Søllested. A sow replacement model using Bayesian updating in a three-level hierarchic Markov process. *Livestock Production Science*, 87(1):25–36, apr 2004.

[10] Sara V. Rodríguez-Sánchez, Lluís M. Plà-Aragonés, and Victor M. Albornoz. Modeling tactical planning decisions through a linear optimization model in sow farms. *Livestock Science*, 143(2-3):162–171, feb 2012.

[11] Sara V. Rodríguez, Victor M. Albornoz, and Lluís M. Plà. A two-stage stochastic programming model for scheduling replacements in sow farms. *TOP*, 17(1):171–189, jul 2009.

[12] Daniel. Babot. *La gestión técnica de las explotaciones porcinas en España*. Ministerio de Medio Ambiente y Medio Rural Marino, Centro de Publicaciones, 2008.

[13] Anders Ringgaard Kristensen and Thomas Algot Søllested. A sow replacement model using bayesian updating in a three-level hierarchic markov process. *Livestock Production Science*, 87(1):13 – 24, 2004.

[14] Pentahoo Kettle. `http://community.pentaho.com/projects/data-integration`. Online; Accessed: May, 2017.

[15] OpenNebula. `http://opennebula.org/`. Online; Accessed: May, 2017.

[16] Stormy. `http://stormy.udl.cat`. Online; Accessed: May, 2017.

[17] Spring. `https://spring.io/`. Online; Accessed: May, 2017.

[18] Maven – Welcome to Apache Maven. `https://maven.apache.org/`. Online; Accessed: May, 2017.

[19] Hibernate. Everything data. `http://hibernate.org/`. Online; Accessed: May, 2017.

[20] MySQL. `http://mysql.com`. Online; Accessed: May, 2017.