

DJANGO ANALYSIS FRAMEWORK TO IMPLEMENT WEB DATABASE APPLICATIONS MARIADB SCRUM AND METHODOLOGY DEVELOPMENT. APPLICATION: WEB APPLICATION FOR MAINTENANCE MECHANIC FOR MEAT INDUSTRIES PUBLIC COMPANY MUNICIPAL SLAUGHTERING AND MEAT PRODUCTS FROM IBARRA

Robinson Danilo CHÁVEZ CABRERA

Career Computer Systems Engineering, IBARRA, IMBABURA

rdchavezc@utn.edu.ec

Abstract. *Mechanical Maintenance System or SIMMEC, for Public Company slaughtering and meat products Ibarra, Django is developed in an open source framework based on an interpreted programming language python; said framework for its characteristics has been used to accelerate the development of a web application for the meat industry. The system works on open source MariaDB a fork of MySQL, a database-compliant relational database with the framework, very popular in our country for its configuration and have a worldwide community support and development; the methodology Scrum is an agile methodology used for a specific task cycles through multidisciplinary teams.*

The results have been satisfactory for the company has since achieved largely solve the problem of power machinery have control routine maintenance, non-routine and corrective; and to record the activities carried out during such maintenance, saving time and avoiding stop the butchering process of the municipal company..

Keywords

Django, MariaDB, Python, Scrum, Mechanical Maintenance, Slaughtering, SIMMEC.

1. Introduction

The science computer have gone evolving with the time and them people have gone modifying its way of addressing a task with technology. The technology market currently puts at our disposal a limited number of teams of all sizes but with robust technical specifications in such a way that a simple team allows us to develop a presentation, write and send mail, write offices, reviewed social networks and we can do everything while we moved to our workplace. Them tools technological us serve to develop our capabilities of exercise of way efficient them activities every day that comply, therefore it solution computer that is you can give to our routine, implies that must search alternatives existing in the marked for meet such activity.

A. Company the company of trail in the city of Ibarra is a public company owned by the municipality of Ibarra, which provides the slaughtering of animals and minor services.

The slaughter of cattle at home has come to be regarded an irregular practice, banned in many countries; the use of machinery in this process has gone improving of such way that can prevent the stress of them animals, keeping thus them components nutritional of the meat. In the company of slaughtering its machinery has fulfilled their life stage and the company has been forced many times to manufacture their own spare parts in order to keep the process of slaughtering, this maintain the quality and hygiene of the meat that would be ready for consumption with a process ordained with the production line. By not having the automation of maintenance that contribute to prevent faults which occur in such a process.

B. Problem

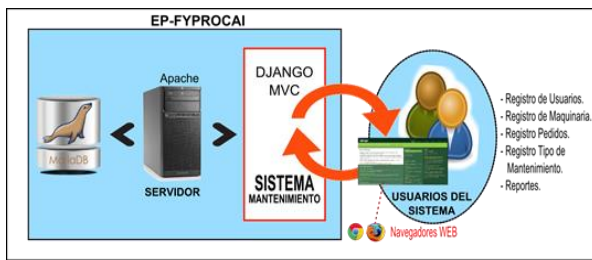
The company public of slaughtering and products meat of Ibarra loses resources to the not count with it automation of them activities of maintenance that facilitates it intervention timely and adequate of the machinery, that is used in the process of slaughtering, because them operators of such machinery report of some damage when such machinery leaves of operate.

C. Objectives document research and the development of the application. Study the language of programming Python. Studying the framework of development of Django. Studying the database RDBMS MariaDB. Determine the benefits and disadvantages to using Django with MariaDB. Study the middleware necessary for the integration of Django and MariaDB.

Apply to the development the methodology SCRUM. Determine the requirements for the implementation of the module.

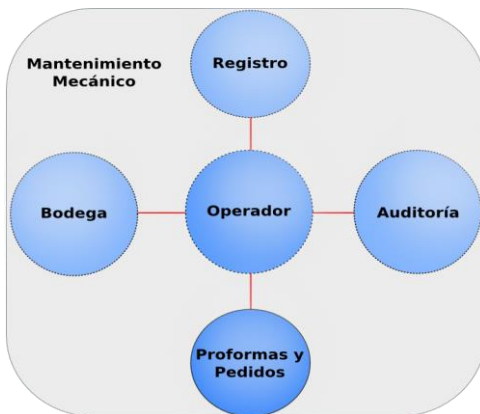
Determine the situation present of them centers of slaughtering focused to the inventory of the machinery. Test the benefits of the technology proposed in the development of an application web of maintenance mechanical for companies that is dedicated to the industrialization meat.

2. The application architecture



In the illustration 1 it indicates its architecture web, and the structure basic functional that will have the system. This is run through a server and then users may access to the system by means of browsers web (Mozilla Firefox, Google Chrome, Opera) installed in your computer from which is may access from different systems operating or distributions GNU / Linux or MS. Windows.

3. Schema of application



A. Register

This thread will record the information required for to perform them schedules of maintenance and of claims of operation of the machinery. The form of income and modification of all teachers to carry out maintenance schedules shall be taken: materials, labour (if necessary) and time of execution of the solution or the work required; These data are updated. Is recorded the plants with which count the company if are separated or if are in the same plant.

B. Proformas and Orders

In this thread is register them proformas chosen for works specialized and that in the workshop not is them can perform, as well as for those orders is will propose a form to request material or accessory of the workshop.

C. Operator

You will be responsible for keep track of machinery and repair of the whole plant is mechanical, electrical or civil;

SIMMEC

with a record of maintenance corrective and preventive with its document of report of status of the maintenance, times of execution and responsible.

D. Storage Cellar

This thread is responsible for providing the information of the existing materials, also tools available and alert when the existing material is almost finished for an upcoming provision.

E. Auditoria

This thread will be responsible for all audit assignment and how much material or what was used prior to approval by the authority.

You will keep control of all changes that are made to the system, data of the users who enter, actions carried out.

This plug-in generates relevant reports for internal audit for review and control.

4. Develop

4.1 Language of programming Python



Ilustración 1: Logo Fuente: python.org

The language for programming excellence of all the people who wish to learn how to program or to develop quality software and feasible since it is cross-platform and easy to understand, since the encoding keeps persistent with iterations, declarations and initializations.

A. What is python?

Language of programming developed by Guido Van Rossum in the end of them 80 and principles of those 90; very similar to the language Perl but more friendly with the developer and with the environment of execution. Language whose main characteristic is an interpreted language or as also is called script that runs with an interpreter and requires no compiler since it does not require be converted to machine language.

B. Characteristics

B.1 Language interpreted or from script

Runs on a non-compiled interpreted environment advantage is its easy portability and flexibility; also possesses the capacity depending on the environment of execution be semi

interpreted as is it can embed with others languages as java, ruby among others.

B.2 Dynamic typing

It feature more debatable by the developers of languages compiled is that not is requires declare variable with the type of data since works with the value that is you assign and is delimited by quotes simple or double, points, brackets, parentheses, keys, classes u others. Typing is not allowed to change the data of the variable which from the beginning was assigned; only fit transformations of integer to string or vice versa if the value is the one who claims to be.

B.3 Multiplatform

To be interpreted or semi-interpretado language execution is more flexible in environments such as UNIX, GNU/Linux, Linux, DOS/Windows, MAC OS, etc. since it offers countless than libraries that systems come pre-configured and Setup is minimal.

B.4 Object Oriented

P.O.O. Is a paradigm of programming that consists in giving solution to a problem from classes and objects.

Python allows interpretative, functional programming and aspect-oriented programming.

```

1 class Fibonacci():
    def __init__(self, num_elementos): 2
        self.num = num_elementos
    3 def get_fibonacci(self):
        x, y = 0, 1
        fibonacci=[x, y]
        4 {
            for n in range(self.num):
                fibonacci.append(x+y)
                aux = x + y
                x = y
                y = aux
            print (fibonacci) 5
valor = Fibonacci(5)
valor.get_fibonacci() 6
    
```

Ilustración 2: Sintaxis Python3 Fuente: Autor

Result:

[0, 1, 1, 2, 3, 5, 8]

Where:

- 1.- Declare object called Fibonacci.
- 2.- Declare the constructor of the object with an attribute.
- 3.- We declare a function of the Fibonacci object with parameter keyword self referencing the same function.
- 4.- We initialized the variables as whole as the collection; We control cycle for calculating the value of the series and we will add it to the array.
- 5.- Print the array with the calculated values of the series.

6.- Instantiate the object in the variable value with the number of cycles of repetition and call the method.

4.2 Framework Django



Ilustración 3: Logo Django Fuente:.djangoproject.com

A. What is a framework?

A framework is a set of tools, artifacts and standards that give functionality extra to an application and facilitates their understanding and development; Therefore these frameworks serve to shorten the time of development and reduce the cost of the investment that this implies; not all developers agree these tools as they called them sources of mediocrity that more experienced developers make their own from scratch and therefore is a source of controversy to this day use or not a framework.

Those frameworks have as objective offer a functionality defined, auto content, being built using patterns of design, and its feature main is its high cohesion and low coupling. For access to that functionality, is built parts, objects, called objects hot that link the needs of the system with the functionality that this provides.

B. History

Django was born in the year 2003 in a medium of communication written Lawrence Journal-World, is developed by Adrian Holovaty and Simmon Willison; in 2005 released to the public the first version of Django under the license BSD; through the Foundation of Software Django (DSF), something restrictive of the license is that is must request authorization to create a fork (fork); its name comes from Jean "Django" Reinhardt jazz guitarist.

C. Development pattern

C.1 Model, this component manages access to data. Is related with the data: how access to it, how validate, how is behaves and the relations between them groups of data.

C.2 Template (template), this "component" is in charge of the presentation; It contains the presentation in form of web page or any other document-related decisions.

C.3 View, this component is a layer of business logic, makes a mediation between models and templates; It contains the logic of the data portion of the model that will be sent to the template (s).

D. Scheme of folders and files in Django

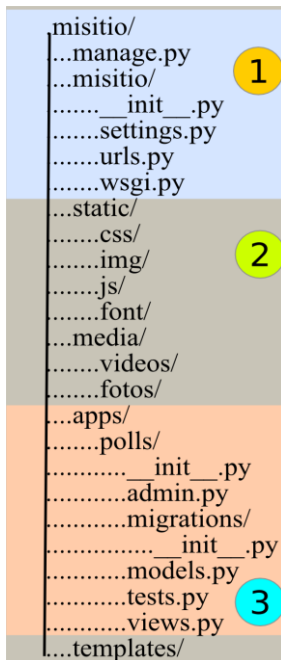


Ilustración 4: Scheme of folders and files of proyecto Django. Fuente: djangoproject.com

Where:

D.1 It is generated when we created the application. In the terminal the command insert:

```
$django-admin starproject mysite
```

The file `__init__.py`, is required for Python recognize this file and the folder that it contains as a package of applications or the project.

The file `manage.py`, for it is used to execute commands in the terminal that affect the project.

The file `settings.py`, of options or settings for this project of Django.

The file `urls.py`, of the declarations of the URLs for a Django project; a "table of contents" (tuple) of your site made with Django. Example:

```
from django.conf.urls import url
from . import views
```

```
urlpatterns = [
    url(r'^$', views.index, name='index'),
]
```

The file `wsgi.py` of server configuration internal wsgi for developing mode and that refers to the direction of the project.

D.2 Scheme that the developer adds in the directories and configuration files.

The `apps/` binder, contains applications that project so many few are necessary; the folder is optional so that the framework does not generate them.

The `static/` binder, the configuration file of the project (`settings.py`) already makes reference to this folder which contains all the static files project is noteworthy that audio and video files, pictures, documents, types of letters, templates, html, etc. are also static that depends on the hierarchy and order that you want to give to the project.

The `media/` binder, is the folder that will contain them files multimedia of the applications that is generated.

The `templates/` binder, is the folder that contains the files of *.html of the application.

D.3 It is generated when you run the command in the terminal:

```
\apps$ python ../manage.py starapp polls
```

Where:

The `__init__.py` files, indicate that they are django packages. The `admin.py` file, records for the django admin models. Example:

```
from django.contrib import admin
from .models import Question
admin.site.register(Question)
```

The directory `migrations/`, is the in engaged of save the script that is running in the database. For example: the `models.py` file, define the attributes of the model to generate the database. Example:

```
class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')
```

The file `test.py`, file archive of the application unit tests. The file `views.py`, is used to store both the business logic functions with views based on classes. Example:

```
from django.http import HttpResponse
def index(request):
    return HttpResponse("Hello, world. You're at the" +
        "polls index.")
```

Files as **forms.py** and **functions.py** added from the schema by default, will be used to improve the appearance of the models in the application template.

4.3 MariaDB database



Ilustración 5: Logo modificado Fuente: mariadb.org, autor

MariaDB, the exit technique of the activism open source; it is a fork of MySQL that recently was part of Sun Microsystems Company which was acquired by Oracle. In 2010. In 2008 Sun Microsystems Acquires MySQL and was made of a large community of developers, then that was acquired by Oracle, the situation changed and currently there are exclusive property of Oracle MySQL plugins for example NoSQL plugin.

The compatibility of the forks depends on the team or group of people who will support and which can drastically alter your code doing this incompatible with its predecessor version.

A. Engine of storage XtraDB

Modified engine that is no more than an improved InnoDB to support "modern" sites which has high availability and flexibility since high-end servers would support taking advantage of this engine, substantially improving its functionality.

B. Compatibility MariaDB and MySQL

In the following table is makes a table comparison proposed by DB-Engines in which proposes some properties.

| Property | MariaDB | MySQL |
|--|---|--------------------------------|
| Description | A fork of MySQL with the objective of offering an improved and the replacement of MySQL developed by the community. | Open source widely used RDBMS. |
| Ranking of storage engines. (Solid IT, 2014) | 27 | 2 |
| scheme guide. (Solid IT, 2014) | 14.734 | 1.309.550 |

| | | |
|---|---|---|
| Web site | http://mariadb.org | http://www.mysql.com |
| Technical documentation | https://mariadb.com/kb/en/mariadb/ | http://dev.mysql.com/doc/ |
| Developer and Support | Monty Program AB | Oracle |
| Release | 2009 | 1995 |
| License | Open Source | Open Source |
| Implementation language | C y C++ | C y C++ |
| Platforms | FreeBSD, Linux, OS X, Solaris and Windows. | FreeBSD, Linux, OS X, Solaris and Windows. |
| Models of databases | Relacional DBMS (with an API to access direct to the motor of storage) | Relacional DBMS (with an API to access direct to the motor of storage) |
| Data schema | Yes | Yes |
| Supports float data type or date | Yes | Yes |
| secondary indexes | Yes | Yes |
| Hold up SQL | Yes (depends of the type of requirement) | Yes (depends of the requirement) |
| API "s and other methods of access | ADO.NET, JDBC y ODBC | ADO.NET, JDBC y ODBC |
| Supports languages of programming | Ada, C, C#, C++, D, Eiffel, Erlang, Haskell, Java, Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme y Tcl. | Ada, C, C#, C++, D, Eiffel, Erlang, Haskell, Java, Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme y Tcl. |
| Procedures stored and/or Triggers | Yes (syntax with some improvements) | Yes |
| It supports methods of partitioning (cluster) | Yes (starting from the 5.4) | Yes |
| Access control | Configurable | Configurable |
| ACID | Yes | Yes |

Tabla 1 Comparison of mariaBD vs MySql version 5.X Fuente: DB-Engines.com

C. Advantages and disadvantages to using Django with MariaDB.

C.1 Advantages

Both Django and MariaDB are Open Source.

Low-cost requirements for the elaboration of databases can be executed on a machine with limited resources without any problem.

Ease of configuration and installation.

Supports wide variety of systems operating.

Low probability of corrupt data, even if those errors not is produced in the own Manager.

Its connectivity and security make of Django and MariaDB highly appropriate for access databases on Internet.

C.2 Disadvantages

The documentation of the API is varied and extensive with regard to MariaDB that with others systems managers of data.

4.4 SCRUM Agile methodology

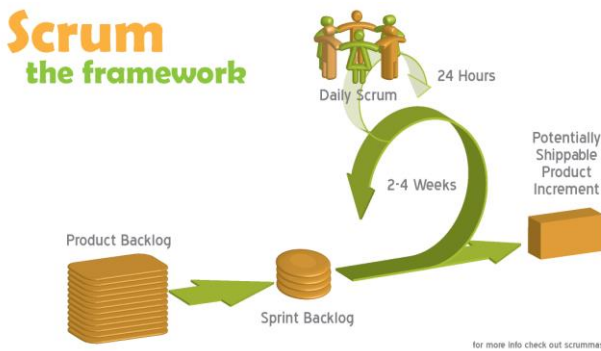


Ilustración 6 Ciclo de vida de SCRUM. Fuente: scrummaster.com.au

It is a guide for the management and development of software based on an iterative and incremental process commonly used in agile software development-based environments.

It can be used in software maintenance teams, or in a program management approach: Scrum of Linde, a development methodology very simple, requiring hard work because it is not based in the monitoring of a plan, but the continuously adapting to the circumstances of the development of the project.

A. Artifacts and events of SCRUM

Stack of product (requirements functional and non functional).

Battery in the sprint (responsibilities).

Sprints.

Graphic of product (Burn Up).

Graph of progress (Burn Down).

Meeting technical (Sprint Planning Meeting).

Meeting of closing sprint and delivery of the increase (Block Log).

B. Vision process

Scrum called "sprint" to each iteration of development and recommended to perform them with durations of 30 days.

The sprint is therefore the central core that provides the basis for iterative and incremental development.

The elements that make up the development Scrum are:

The meetings

Sprint planning: prior to the start of each sprint in which determines what will be the work and the objectives that must be met in that iteration.

Meeting daily: brief review of the team of the work performed to the date and the forecast for the day following.

Review of sprint: analysis and review of the increment generated.

Values

Scrum is a help to organize to the people and the flow of work; as it can be other proposals of forms of work agile.

Delegation of powers (empowerment) to the team that can auto-organize is and take the decisions on the development.

Respeto entre las personas. Los miembros del equipo deben confiar entre ellos y respetar sus conocimientos y capacidades.

Responsibility and self-discipline (not discipline imposed). Work focused on the development of the committed.

Information, transparency and visibility of the development of the project.

5. Design and application development and implementation.

A. Description General the methodology

A.1 Members and roles of the project.

| Member | Contact | Role |
|----------------|-----------|-----------------------------|
| Kléver Taboada | 62546230 | Coordinator / Scrum Manager |
| Luis Ramírez | 62546230 | Technical |
| Xavier Rea | 986099536 | Adviser |
| Danilo Chávez | 989313653 | Developer |

Tabla 2 Team SCRUM Fuente: Autor

A.3 Sprint Backlog

| SPRINT | | HICHO | | DURACIÓN | | Tareas pendientes | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------------|-----------------------------------|------------|-----------|--------------------|----|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| PILA DEL SPRINT | | Tarea | | Estado - Responsal | | EFUEZRO | | | | | | | | | | | | | | | | | | | | | | |
| Tareas pendientes | | | | | | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| Horas de trabajo pendientes | | | | | | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| Backlog | Tarea | Estado | Responsal | Esfuerzo | | | | | | | | | | | | | | | | | | | | | | | | |
| Inicio | Establecer acciones | Analisis | Yover | 16 | 16 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| I1 | Establecer procesos | Procesado | Dante | 8 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I2 | Establecer Estrategias | Planteado | Dante | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I3 | Establecer Software Grafico | Codificado | Dante | 4 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I4 | Establecer calendario | Codificado | Yover | 16 | 16 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I5 | Instalar dictio Debian | Planteado | Dante | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I6 | Instalar Python | Codificado | Dante | 8 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I7 | Instalar modulares | Codificado | Dante | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I8 | Instalar Django | Planteado | Dante | 8 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I9 | Instalar Base de datos | Planteado | Dante | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I10 | Correccion de bugs de instalacion | Codificado | Dante | 8 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I11 | Instalar modulares 1 | Codificado | Dante | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I12 | Modelo Entidad Relacion | Codificado | Dante | 8 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I13 | Modelo Pruebas | Codificado | Dante | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I14 | Modelo Frontend | Codificado | Dante | 8 | 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| I15 | Respaldo | Codificado | Dante | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | |
| Segundo Registro | | | | Esfuerzo | | | | | | | | | | | | | | | | | | | | | | | | |
| S1 | Desarrollo de las interfaces | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| S2 | Desarrollo de los modelos | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| S3 | Desarrollo de las vistas | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| S4 | Desarrollo de los controladores | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| S5 | Configuracion URLS | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| Tercero Registro | | | | Esfuerzo | | | | | | | | | | | | | | | | | | | | | | | | |
| T1 | Desarrollo de las interfaces | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| T2 | Desarrollo de los modelos | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| T3 | Desarrollo de las vistas | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| T4 | Desarrollo de los controladores | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| T5 | Configuracion URLS | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| Cuarto Registro y Párrafos | | | | Esfuerzo | | | | | | | | | | | | | | | | | | | | | | | | |
| C1 | Desarrollo de las interfaces | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| C2 | Desarrollo de los modelos | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| C3 | Desarrollo de las vistas | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| C4 | Desarrollo de los controladores | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| C5 | Configuracion URLS | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| Quinto Registro | | | | Esfuerzo | | | | | | | | | | | | | | | | | | | | | | | | |
| Q1 | Desarrollo de las interfaces | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| Q2 | Desarrollo de los modelos | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| Q3 | Desarrollo de las vistas | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| Q4 | Desarrollo de los controladores | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| Q5 | Configuracion URLS | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| Sexto Registro | | | | Esfuerzo | | | | | | | | | | | | | | | | | | | | | | | | |
| S6 | Desarrollo de las interfaces | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| S7 | Desarrollo de los modelos | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| S8 | Desarrollo de las vistas | Codificado | Dante | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| S9 | Desarrollo de los controladores | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| S10 | Configuracion URLS | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| Septimo Documentacion | | | | Esfuerzo | | | | | | | | | | | | | | | | | | | | | | | | |
| D1 | Elaboracion del documento | Codificado | Dante | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |

Ilustración 7 Sprint Blocklog Fuente: Autor

6. Conclusions

The work of this thesis related with the development of software contribute in the automation of processes of the different industries or companies where is developed.

It develop with the language python allows to them programmers maintain an order and an ethics of programming, this will achieve applications more clean and with the lower number of lines of code possible since has not is repeated code.

The Django framework accelerates the development of applications since it automatically generates ready to use code.

Django is a framework that is won followers and maintains a community that is kept constantly updated since python is a language known for free software developers.

The relational data base MariaDB is a robust relational database that supports the load of the framework since it is a fork of MySQL, a database recognized worldwide.

The methodology used in this project was SCRUM allowed a development of quality and with the fewest errors, so it was necessary to have a fully identified development team knowing the strengths of each of the members.

SCRUM is recommended for projects medium and small where it required a solution agile and of results visible in short time.

He Software free us helps to reduce notably them costs in the development of a project, since the developer or the team not have that verse affected by the purchase of licenses of use that in many cases is high.

To the keep them processes automated is ensures that the service of this company is going to improve significantly.

Bibliographic References

- [1] Alchin, M. (2013). *Pro Django, Second Edition*. LA California: Apress.
- [2] Apache Org. (29 de Agosto de 2014). *FAQ Apache Org*. Obtenido de <http://www.apache.org/foundation/preFAQ.html>
- [3] Bennett, J. (2009). *Practical Django Projects Second Edition*. New York, UE: Duncan Parkes.
- [4] Briggs, J. R. (2007). *Snake Wrangling for Kids, Learning to Program with Python*. San Francisco, California.: Creative Commons 3.0 España.
- [5] Debian. (2013). *Debian*. Obtenido de Caracteristicas: <https://www.debian.org/intro/about>
- [6] Downey, A., Elkner, J., & Meyers, C. (2002). *Aprenda a Pensar Como un Programador con Python*. Wellesley, Massachusetts: Green Tea Press.
- [7] Empresa Pública Metropolitana de Rastro. (28 de Octubre de 2014). *Empresa Pública Metropolitana de Rastro Quito*. Obtenido de <http://www.epmrq.gob.ec>
- [8] Fab Force. (2014). *Dbdesigner4*. Obtenido de <http://www.fabforce.net/dbdesigner4/>
- [9] Follet, M. (24 de Noviembre de 2014). *Soy Admin*. Obtenido de Instalar MariaDB en CentOS con Mysql en producción: <http://soyadmin.com/2014/11/instalar-mariadb-en-centos-con-mysql-en-produccion/>
- [10] Forta, B. (2012). *MariaDB Crash Course*. Crawfordsville, Indiana: Addison Wesley.
- [11] Holovaty Adrian, K.-M. J. (2007). *El libro de Django 1.0*. Obtenido de Libros Web: http://librosweb.es/libro/django_1_0/capitulo_19/exposicion_de_mensajes_de_error.html
- [12] Hundhausen, R. (2012). *Professional Scrum Development with Microsoft® Visual Studio® 2012*. Redmond, Washington: Christian Holdener, S4 Carlisle Publishing Services.
- [13] Janssen, C. (s.f.). *ACID*. Obtenido de Atomicity Consistency Isolation Durability (ACID): <http://www.techopedia.com/definition/23949/atomicity-consistency-isolation-durability-acid>
- [14] MariaDB. (Septiembre de 2015). *MariaDB versus MySQL - Features*. Obtenido de <https://mariadb.com/kb/en/mariadb/mariadb-vs-mysql-features/>

- [15] MariaDB Org. (Septiembre de 2015). *MariaDB Knowledge Base*. Obtenido de <https://mariadb.com/kb/en/>
- [16] Mneptok. (20 de Julio de 2010). Obtenido de Rename Maria Contest Winner: <http://blogs.gnome.org/mneptok/2010/07/20/rename-maria-contest-winner/>
- [17] Palacio, J. (2007). *Flexibilidad con Scrum*. SafeCreative.
- [18] PhpMyAdmin . (2013). *PhpMyAdmin - Features*. Obtenido de <https://www.phpmyadmin.net/>
- [19] Q7 Enterprises, Inc. (20 de Junio de 2014). *Manifesto for Agile Software Development* . Obtenido de <http://www.agilemanifesto.org/iso/es/principles.html>
- [20] Rossum, G. v. (2009). *El tutorial de Python (Traducción)*. Argentina: Fred L. Drake, Jr. Python Software Foundation.
- [21] Solid IT. (2014). *DB.Engine*. Obtenido de <http://db-engines.com/en/system/MariaDB>
- [22] Soto, J. (Agosto de 2013). *Tux Apuntes*. Obtenido de Conociendo un Poco de MariaDB: <http://tuxapuntes.com/2013/08/conociendo-un-poco-de-mariadb/>
- [23] Stroganov, A. (29 de 01 de 2014). *Percona*. Obtenido de Percona Server: Thread Pool Improvements for Transactional Workloads: <http://www.percona.com/blog/2014/01/29/percona-server-thread-pool-improvements/>
- [24] Turbogears.org. (Junio de 2010). *TurboGears*. Obtenido de About: <http://www.turbogears.org>
- [25] Widenius, M. ". (2010). *Storage Engines*. Obtenido de Information on Storage Engines included in MariaDB.: <https://mariadb.com/kb/en/mariadb/documentation/s-storage-engines/>
- [26] Widenius, M. (2013). *About MariaDB*. Obtenido de Mariadb.org: <https://mariadb.com/kb/en/>

About the Authors...

Robinson Danilo CHÁVEZ CABRERA, convinced that the technology we will help to solve the greater amount of problems, mostly dysfunction the health of the people. Member of the Association of free Software based Ibarra. 2014 FLISOL Ibarra 2014 organization. Responsible technician of the project cities digital of the GAD Antonio before 2015.