ANÁLISIS DEL FRAMEWORK DJANGO PARA IMPLEMENTAR APLICACIONES WEB CON BASE DE DATOS MARIADB Y METODOLOGÍA DE DESARROLLO SCRUM. APLICATIVO: APLICACIÓN WEB PARA MANTENIMIENTO MECÁNICO EN INDUSTRIAS CÁRNICAS PARA LA EMPRESA PÚBLICA MUNICIPAL DE FAENAMIENTO Y PRODUCTOS CÁRNICOS DE IBARRA

Robinson Danilo CHÁVEZ CABRERA

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES, IBARRA, IMBABURA

rdchavezc@utn.edu.ec

Resumen. *El objetivo principal de este proyecto de tesis es el desarrollo del Sistema de Mantenimiento Mecánico o SIMMEC, para la Empresa Pública de Faenamiento y Productos Cárnicos de Ibarra, que está desarrollado en Django, el cual es un framework open source basado en el lenguaje programación interpretado python; dicho framework por sus características ha sido utilizado para agilizar el desarrollo de una aplicación web para la industria cárnica. El sistema trabaja sobre MariaDB un fork open source de MySQL, una base de datos relacional compatible con el framework, muy popular en nuestro medio por su configuración y por contar con una comunidad a nivel mundial de soporte como de desarrollo; la metodología utilizada Scrum es una metodología ágil que utiliza ciclos para una tarea específica por medio de equipos multidisciplinarios.*

Los resultados obtenidos han sido satisfactorios para la empresa ya que ha logrado en gran medida solucionar el problema de la maquinaria al poder tener un control del mantenimiento rutinario, no rutinario y correctivo; así como poder registrar las actividades a realizarse durante dicho mantenimiento, ahorrando tiempo y evitando detener el proceso de faenamiento de la empresa municipal.

Palabras Claves

Django, MariaDB, Python, Scrum, Mantenimiento Mecánico, Faenamiento, SIMMEC.

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. Introducción

Las ciencias informáticas han ido evolucionando con el tiempo y las personas han ido modificando su manera de abordar una tarea con tecnología. El mercado tecnológico actualmente pone a nuestro alcance un número limitado de equipos de todos los tamaños pero con especificaciones técnicas robustas de tal manera que un simple equipo nos permita desarrollar una presentación, redactar y enviar correo, escribir oficios, revisamos redes sociales y todo lo podemos hacer mientras nos trasladamos a nuestro lugar de trabajo.

Las herramientas tecnológicas nos sirven para desarrollar nuestras capacidades de ejercer de manera eficiente las actividades cotidianas que cumplimos, por lo tanto la solución informática que se le pueda dar a nuestra rutina, implica que debemos buscar alternativas existentes en el marcado para cumplir dicha actividad.

A. Empresa

La Empresa de Rastro de la ciudad de Ibarra es una empresa de carácter público administrada por el Municipio de Ibarra que brinda los servicios de faenamiento de animales mayores y menores.

El sacrificio de ganado en el domicilio ha pasado a ser considerado una práctica irregular, prohibida en muchos países; la utilización de maquinaria en este proceso ha ido mejorando de tal manera que pueda evitar el estrés de los animales, conservando así los componentes nutricionales de la carne.

En la empresa de faenamiento su maquinaria ha cumplido su etapa de vida útil y la empresa se ha visto obligada muchas veces a fabricar sus propios repuestos con el fin de mantener el proceso de faena, con esto mantener la calidad e higiene de la carne que estaría lista para el consumo con un proceso ordenado con la cadena de producción. Al no contar con la automatización del mantenimiento que coadyuven a evitar los desperfectos que se ocasionan en dicho proceso.

B. Problema

La Empresa Pública de Faenamiento y Productos Cárnicos de Ibarra pierde recursos al no contar con la automatización de las actividades de mantenimiento que facilite la intervención oportuna y adecuada de la maquinaria, que es utilizada en el proceso de faenamiento, por cuanto los operadores de dicha maquinaria informan de algún desperfecto cuando dicha maquinaria deja de funcionar.

C. Objetivos

Documentar la investigación y el desarrollo del aplicativo.

Estudiar el lenguaje de programación Python.

Estudiar el marco de desarrollo de Django.

Estudiar la base de datos RDBMS MariaDB.

Determinar los beneficios y desventajas al utilizar Django con MariaDB.

Estudiar los middlewares necesarios para la integración de Django y MariaDB.

Aplicar al desarrollo la metodología SCRUM.

Determinar los requerimientos para la implementación del módulo.

Determinar la situación actual de los centros de faenamiento enfocado al inventario de la maquinaria.

Probar las bondades de la tecnología propuesta en el desarrollo de una aplicación web de mantenimiento mecánico para empresas que se dedican a la industrialización cárnica.

2. Arquitectura de la aplicación

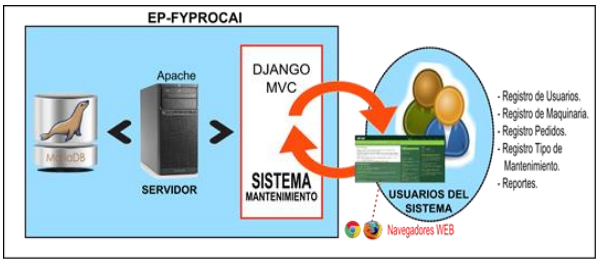


Ilustración 1: Arquitectura a usarse. Fuente: Autor

En la ilustración 1 se indica la arquitectura web, y la estructura básica funcional que tendrá el sistema. Este se ejecutará a través de un servidor y los usuarios podrán acceder al sistema por medio de navegadores web (Mozilla Firefox, Google Chrome, Opera) instalado en su Computador desde la cual se podrá acceder desde diferentes sistemas operativos o distribuciones GNU/Linux o MS. Windows.

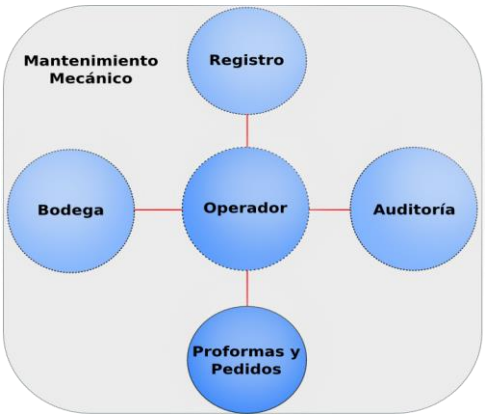


Ilustración 2: Subprocesos del módulo Mantenimie
Fuente: Autor

3. Esquema del aplicativo

A. Registro

Este subproceso registrará la información necesaria para poder realizar los cronogramas de mantenimientos y de reclamos de funcionamiento de la maquinaria.

Se tendrán los formulario de ingresos y modificación de todos los maestros para la realización de los cronogramas de mantenimiento: materiales utilizados, mano de obra (si fue necesario) y tiempos de ejecución de la solución o del trabajo requerido; estos datos se actualizan.

Se registrarán las plantas con las que cuente la empresa si están separadas o si están en la misma planta.

B. Proformas y Pedidos

En este subproceso se registraran las proformas solicitadas para trabajos especializados y que en el taller no se las puede realizar, así como para los pedidos se propondrá un formulario para solicitar material o accesorio del taller.

C. Operador

Será encargado de mantener el historial de la maquinaria y reparaciones de toda la planta sea mecánica, eléctrica o civil; con un registro de mantenimientos correctivo y preventivo con su documento de informe de estado del mantenimiento, tiempos de ejecución y responsable.

D. Bodega

Este subproceso se encargará de proporcionar la información del existente de materiales, también el de herramientas disponibles y alertar cuando el existente de material esté por terminar para una próxima provisión.

E. Auditoría

Este subproceso se encargará de realizar toda la auditoría de asignación y cuanto material o para qué fue utilizado previo a la autorización de la autoridad.

Se mantendrá un control de todos los cambios que se realicen en el sistema, datos del usuarios que ingresa, que acciones realizó.

Este módulo generará los informes pertinentes para auditoría interna para su revisión y control.

4. Desarrollo

4.1 Lenguaje de programacón Python

Ilustración 3: Logo Fuente: python.org

El lenguaje para programación por excelencia de todos las personas que deseen aprender a programar o desarrollar software de calidad y factible ya que es multiplataforma y fácil de comprender, ya que mantiene persistente la codificación con las iteraciones, declaraciones e inicializaciones.

A. ¿Qué es Python?

Lenguaje de programación desarrollado por Guido Van Rossum en los finales de los 80 y principios de los 90; muy parecido al lenguaje Perl pero más amigable con el desarrollador y con el entorno de ejecución. Lenguaje cuya característica principal es que es un lenguaje interpretado o de como también se lo llama script que se ejecuta con un intérprete y no requiere de compilador ya que no requiere de ser transformado a lenguaje máquina.

B. Características

B.1 Lenguaje interpretado o de script

Se ejecuta en un entorno interpretado no compilado la ventaja es su fácil portabilidad y flexibilidad; también posee la capacidad dependiendo del entorno de ejecución ser semi interpretado ya que se lo puede embeber con otros lenguajes como java, ruby entre otros.

B.2 Tipado dinámico

La característica más debatible por los desarrolladores de lenguajes compilados es que no se requiere declarar variables con el tipo de dato ya que trabaja con el valor que se le asigne y es delimitado por comillas simples o dobles, puntos, corchetes, paréntesis, llaves, clases u otras.

El tipado no permite cambiar el dato de la variable que desde un comienzo se le asignó; solo caben las transformaciones propias de enteros a cadena o viceversa si el valor es el que dice ser.

B.3 Multiplataforma

Al ser un lenguaje interpretado o semi-interpretado su ejecución es más flexible en entornos como UNIX, GNU/Linux, Linux, DOS/Windows, MAC OS, etc. ya que ofrece un sinnúmero que librerías que los sistemas vienen pre-configurados y la configuración es mínima.

B.4 Orientado a Objetos

P.O.O. Es un paradigma de programación que consiste en dar solución a un problema desde clases y objetos.

Python permite programación interpretativa, funcional, y programación orientada a aspectos.

B.5 Sintaxis

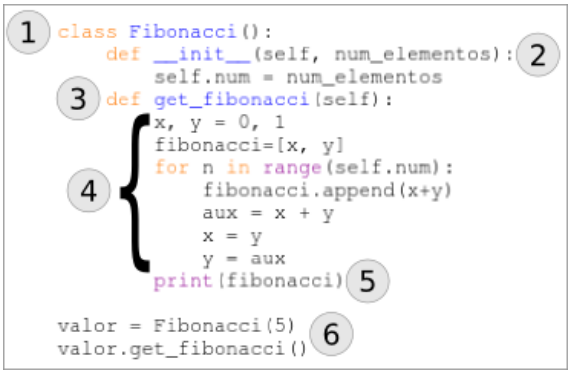


Ilustración 4: Sintaxis Python3 Fuente: Autor

Resultado:

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

Dónde:

1.- Declaramos objeto llamado Fibonacci.

2.- Declaramos el constructor del objeto con un atributo.

3.- Declaramos una función del objeto *Fibonacci* con parámetro la palabra reservada *self* que hace referencia a la misma función.

4.- Inicializamos las variables tanto enteras como la colección; y, controlamos con un ciclo *for* calcular el valor de la serie y lo vamos añadiendo al array.

5.- Imprimir el array con los valores calculados de la serie.

6.- Instanciar el objeto en la variable *valor* con el número de ciclos de repetición y llamamos el método.

4.2 Framework Django

Ilustración 5: Logo Django Fuente: djangoproject.com

A. ¿Qué es un framework?

Un framework es un conjunto de herramientas, artefactos y estándares que dan funcionalidad extra a una aplicación y facilita su comprensión y desarrollo; por lo tanto estos frameworks sirven para acortar el tiempo de desarrollo y reducir el costo de la inversión que esto implica; no todos los desarrolladores están de acuerdo con estas herramientas ya que las llaman fuentes de mediocridad por que los programadores más experimentados los hacen ellos solos desde cero y por lo tanto es un motivo de polémica hasta hoy en día el uso o no de un framework.

Los frameworks tienen como objetivo ofrecer una funcionalidad definida, auto contenido, siendo construidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento. Para acceder a esa funcionalidad, se construyen piezas, objetos, llamados objetos calientes, que vinculan las necesidades del sistema con la funcionalidad que este presta.

B. Historia

Django nace en el año 2003 en un medio de comunicación escrita Lawrence Journal-World, es desarrollado por Adrián Holovaty y Simmon Willison; en 2005 lanzaron al público la primera versión de Django bajo la licencia BSD; mediante la Fundación de Software Django (DSF), algo restrictivo de la licencia es que se debe pedir autorización para crear una bifurcación (fork); su nombre proviene del guitarrista de jazz Jean "Django" Reinhardt.

C. Patrón de desarrollo

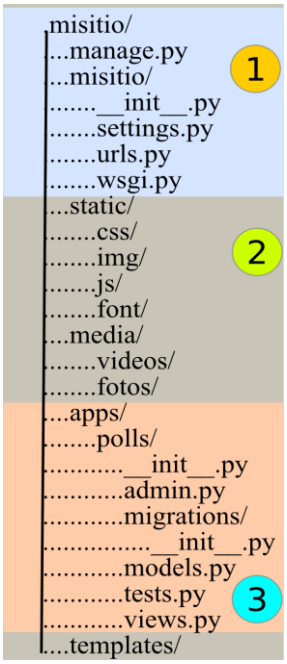
C.1 Modelo, este componente se encarga de gestionar el acceso a datos. Está relacionado con los datos: cómo acceder a ella, cómo validar, cómo se comporta y las relaciones entre los grupos de datos.

C.2 Template (Plantilla), este "componente" se encarga de la presentación; contiene las decisiones relacionadas con la presentación en forma de página web o algún otro documento.

C.3 Vista, este componente es una capa de lógica del negocio, hace una intermediación entre los modelos y las plantillas; contiene la lógica de la porción de datos del modelo que serán enviadas a la plantilla(s).

D. Esquema de carpetas y archivos en Django

Ilustración 6: Esquema de carpetas y archivos de proyecto Django. Fuente: djangoproject.com



Dónde:

D.1 Se genera cuando creamos la aplicación. En el terminal insertamos el comando:

```
$django-admin starpoject mysite
```

El archivo ***__init__.py***, es requerido para que Python reconozca este archivo y la carpeta que lo contiene como un paquete de aplicaciones o el proyecto.
El archivo ***manage.py***, para se lo utiliza para ejecutar comandos en la terminal que afectan al proyecto.
El archivo ***settings.py***, de opciones y/o configuraciones para este proyecto de Django.
El archivo ***urls.py***, de las declaraciones de las direcciones URL para un proyecto de Django; una "tabla de contenidos"(tupla) de tu sitio hecho con Django. Ejemplo:

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

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

El archivo ***wsgi.py***, de configuración del servidor wsgi interno para el modo en desarrollo y que hace referencia a la dirección del proyecto.

D.2 Esquema que el desarrollador agrega en el directorios y archivos de configuración.

La carpeta **apps/**, contiene las aplicaciones de ese proyecto tantas cuantas sean necesarias; la carpeta es opcional por eso que el framework no las genera.
La carpeta ***static/,*** el archivo de configuración del proyecto (settings.py) ya hace referencia a esta carpeta que contendrá todos los archivos estáticos del proyecto cabe recalcar que los archivos de audio y vídeo, fotos, documentos, tipos de letras, plantillas html, etc. son también estáticos eso depende de la jerarquía y orden que se le quiera dar al proyecto.
La carpeta ***media/***, es la carpeta que contendrá los archivos multimedia de las aplicaciones que se generen.
La carpeta ***templates/***, es la carpeta que contiene los archivos de *.html de la aplicación.

D.3 Se genera cuando en el terminal ejecutamos el comando:

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

Dónde:

Los archivos *__init__.py*, indican que son paquetes de django.
El archivo *admin.py*, registra los modelos para el admin de django. Ejemplo:

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

El directorio *migrations/*, es el en ecargado de guardar los script que se ejecutan en la base de datos. Por ejemplo:
El archivo *models.py*, definimos los atributos del modelo a generar en la base de datos. Ejemplo:

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

El archivo *test.py*, archivo de pruebas unitarias de la aplicación.
El archivo *views.py*, archivo que es utilizado para guardar la lógica de negocio tanto de funciones con de vistas basadas en clases. Ejemplo:

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

Los archivos como *forms.py* y *functions.py* que se añaden a partir del esquema por defecto, servirán para mejorar el aspecto de los modelos en el template de la aplicación.

4.3 Base de datos MariaDB

Ilustración 7: Logo modificado Fuente: mariadb.org, autor

MariaDB, la salida técnica del activismo open source; es un fork de MySQL que hace poco era parte de Sun Microsystems empresa que fue adquirida por Oracle. En 2010. En 2008 Sun Microsystems adquiere MySQL y se hizo de una gran comunidad de desarrolladores, luego que fue adquirida por Oracle la situación cambió y actualmente hay plugins de MySQL de propiedad exclusiva de Oracle por ejemplo el plugin de NoSQL.

La compatibilidad de los forks depende del equipo o grupo de personas que darán soporte y que pueden alterar

drásticamente su código haciendo de esta versión incompatible con su predecesor.

A. Motor de almacenamiento XtraDB

El modificado motor que no es más que un InnoDB mejorado para soportar sitios "modernos" que tiene alta disponibilidad y flexibilidad ya que soportaría servidores de alta gama sacándole el máximo provecho a dicho motor, mejorando sustancialmente su funcionalidad.

B. Compatibilidad MariaDB y MySQL

En la siguiente tabla se hace una tabla comparativa propuesta por DB-Engines en la que propone algunas propiedades.

Propiedad	MariaDB	MySQL
Descripción	Un fork de MySQL con el objetivo de ofrecer un mejorado y el reemplazo de MySQL desarrollada por la comunidad.	De código abierto ampliamente usada RDBMS.
Ranking de los motores de almacenamiento. (Solid IT, 2014)	27	2
Esquema de guía. (Solid IT, 2014)	14.734	1.309.550
Sitio WEB	http://mariadb.org	http://www.mysql.com
Documentación Técnica	https://mariadb.com/kb/en/mariadb/	http://dev.mysql.com/doc/
Desarrollador y Soporte	Monty Program AB	Oracle
Año de lanzamiento	2009	1995
Tipo Licencia	Open Source	Open Source
Lenguaje de implementación	C y C++	C y C++
Plataformas	FreeBSD, Linux, OS X, Solaris y Windows.	FreeBSD, Linux, OS X, Solaris y Windows.
Modelo de base de datos	Relacional DBMS (con un API para acceder directo al motor de almacenamiento)	Relacional DBMS (con un API para acceder directo al motor de almacenamiento)
Esquema de datos	Si	Si
Soporta tipos de datos float o date	Si	Si
Índices secundarios	Si	Si
Soporta SQL	Si (depende del tipo de requerimiento)	Si (depende del requerimiento)
API"s y otros métodos de conexión y acceso	ADO.NET, JDBC y ODBC	ADO.NET, JDBC y ODBC
Soporta lenguajes de programación	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.
Procedimientos almacenados y/o Triggers	Si (sintaxis con algunas mejoras)	Si
Soporta Métodos de particionamiento (clúster)	Si (a partir de la 5.4)	Si
Control de acceso	Configurable	Configurable
ACID	Si	Si

Tabla 1Comparación de mariaBD vs MySql versión 5.X Fuente: DB-Engines.com

C. Ventajas y desventajas al utilizar Django con MariaDB.

C.1 Ventajas

Tanto Django como MariaDB son Open Source.

Bajo costo en requerimientos para la elaboración de bases de datos puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

Facilidad de configuración e instalación.

Soporta gran variedad de Sistemas Operativos.

Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor.

Su conectividad y seguridad hacen de Django y MariaDB altamente apropiado para acceder bases de datos en Internet.

C.2 Desventajas

La documentación de las API es variada y extensa con respecto a MariaDB que con otros sistemas gestores de datos.

4.4 Metodología ágil SCRUM

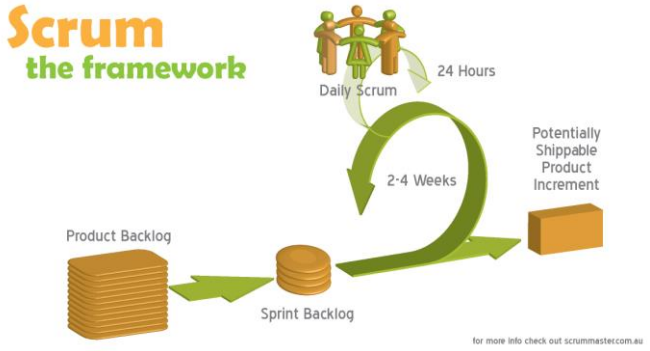


Ilustración 8 Ciclo de vida de SCRUM.Fuente: scrummaster.com.au

Es una guía para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados para el desarrollo ágil de software.

Puede ser utilizada en equipos de mantenimiento de software, o en una aproximación de gestión de programas: Scrum de Scrums, una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

A. Artefactos y eventos de SCRUM

Pila de producto (requerimientos funcionales y no funcionales).

Pila del sprint (Responsabilidades).

Sprints.
Gráfica de producto (Burn Up).
Gráfica de avance (Burn Down).
Reunión técnica (Sprint Planning Meeting).
Reunión de cierre de sprint y entrega del incremento (Block Log).

B. Visión del proceso

Scrum denomina "sprint" a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días. El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental.

Los elementos que conforman el desarrollo Scrum son:

Las reuniones

Planificación de sprint: Previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.

Reunión diaria: Breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.

Revisión de sprint: Análisis y revisión del incremento generado.

Valores

Scrum es una ayuda para organizar a las personas y el flujo de trabajo; como lo pueden ser otras propuestas de formas de trabajo ágil.

Delegación de atribuciones (empowerment) al equipo para que pueda auto-organizarse y tomar las decisiones sobre el desarrollo.

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

Responsabilidad y auto-disciplina (no disciplina impuesta).

Trabajo centrado en el desarrollo de lo comprometido.

Información, transparencia y visibilidad del desarrollo del proyecto.

5. Diseño y desarrollo e implementación del aplicativo.

A. Descripción General de la Metodología

A.1 Integrantes y roles del proyecto.

Integrante	Contacto	Rol
Kléver Taboada	62546230	Coordinador/Scrum Manager
Luis Ramírez	62546230	Técnico
Xavier Rea	986099536	Asesor
Danilo Chávez	989313653	Desarrollador

Tabla 2Equipo SCRUM Fuente: Autor

A.3 Pila del Sprint

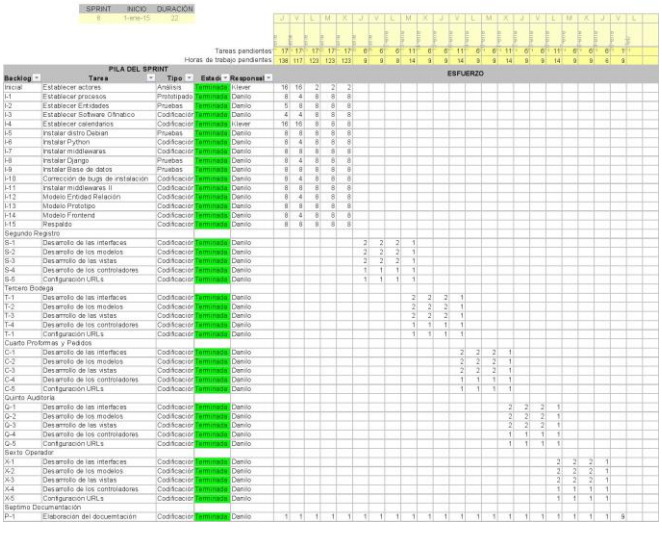


Ilustración 9 Sprint Blocklog Fuente: Autor

6. Conclusiones

Los trabajos de tesis relacionados con el desarrollo de software contribuyen en la automatización de procesos de las diferentes industrias o empresas donde se desarrollan.

El desarrollar con el leguaje python permite a los programadores mantener un orden y una ética de programación, esto logrará aplicaciones más limpias y con el menor número de líneas de código posible ya que tiene no se repite código.

El framework Django acelera el desarrollo de aplicaciones debido a que genera automáticamente código listo para ser usado.

Django es un framework que va ganado adeptos y mantiene una comunidad que se mantiene constante actualización ya que python es un lenguaje muy conocido por los desarrolladores de software libre.

La base de datos relacional MariaDB es una base de datos relacional robusta que soporta la carga del framework ya que es un fork de MySQL, una base de datos reconocida mundialmente.

La metodología de desarrollo usada en este proyecto fue SCRUM permitió un desarrollo de calidad y con el menor número de errores, para ello fue necesario contar con un equipo de desarrollo plenamente identificado conociendo los puntos fuertes de cada uno de los integrantes.

SCRUM es recomendable para proyectos medianos y pequeños donde se requiera una solución ágil y de resultados visibles en corto tiempo.

El Software Libre nos ayuda a reducir notablemente los costos en el desarrollo de un proyecto, ya que el desarrollador o el equipo no tendrán que verse afectado por la compra de licencias de uso que en muchos casos es elevada.

AL mantener los procesos automatizados se asegura que el servicio de dicha empresa va a mejorar notablemente.

Referencias Bibliográficas

[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/storage-engines/

[26] Widenius, M. (2013). *About MariaDB*. Obtenido de Mariadb.org: https://mariadb.com/kb/en/

Sobre los Autores...

Robinson Danilo CHÁVEZ CABRERA, convencido que la tecnología nos ayudará a solucionar la mayor cantidad de problemas, sobre todo precautelar la salud de las personas. Miembro de la Asociación de Software Libre sede Ibarra. 2014. Organización de FLISOL Ibarra 2014. Responsable técnico del Proyecto Ciudades Digitales del GAD Antonio Ante 2015.